LIBRARY OF THE
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

510.84
Il6r
no. 590-594
cop. 2



IIR GN
no.594
cop. 2

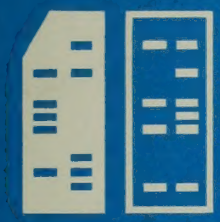
Report No. UIUCDCS-R-73-594

ARRAYS IN PL/I
A LESSON FOR PLATO IV

by

Luis Humberto Jáuregui-Topete

October 1973



DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

THE LIBRARY OF THE
NOV 2 8 1973
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

Report No. UIUCDCS-R-73-594

ARRAYS IN PL/I
A LESSON FOR PLATO IV*

by

Luis Humberto Jáuregui-Topete

October 1973

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

*This work was supported in part by the Department of Computer Science and submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science, October 1973.



Digitized by the Internet Archive
in 2013

http://archive.org/details/arraysinplilesso594jure

ACKNOWLEDGMENT

The author wishes to express his gratitude to Professor H. George Friedman, Jr., for his direction, comments, suggestions, and his valuable time spent at PLATO IV reviewing the lesson.

I also wish to express my appreciation to my wife for typing the manuscript and for her encouragement.

TABLE OF CONTENTS

		Page
1.	INTRODUCTION	1
1.1	Automation of Computer Science 100-Level Courses	1
1.2	The PLATO IV System	1
1.3	Lessons in PLATO IV	2
2.	AUTOMATION OF C.S. 100-LEVEL COURSES	3
2.1	Library of Lessons	3
2.2	The GUIDE	3
2.3	Lessons on PL/I	4
2.4	PL/I Sequence	4
2.5	Intended Audience of the PL/I Sequence	4
2.6	Prerequisites	5
3.	ARRAYS IN PL/I	6
3.1	Purpose of the Lesson	6
3.2	Development of the Lesson	6
3.3	Trial with Students	8
3.4	Future of the Lesson	9
LIST OF REFERENCES		11
APPENDIX -- TUTOR PROGRAM "ARRAYS IN PL/I"		12

1. INTRODUCTION

1.1 Automation of Computer Science 100-Level Courses

The Department of Computer Science of the University of Illinois has started a project to automate all computer science courses in the 100 level, coordinated by Professor Jurg Nievergelt. The automation is being done in PLATO IV, a system designed to use the capabilities of a modern digital computer as an aid for instruction.

1.2 The PLATO IV System

PLATO IV is a real time computer system developed at the University of Illinois, composed of a computer and several hundred terminals. The terminals, connected by a direct line or by a telephone call to the computer, may be located anywhere around the world. For details on the PLATO IV system see reference [1].

The terminals normally have a screen for computer output and a keyboard for student input. They can also have a special screen that accepts touching as the student's response, and an audio device for the computer to send audio output to the student. The square screen (plasma panel), developed at the University of Illinois, has a matrix of 512 × 512 dots which can be lit individually, giving a resolution better than that of commercial television. Text, as well as figures and special characters, can be shown on the screen.

The total design of the system allows for interaction with the student on a one-to-one basis. PLATO responds quickly to the student input on the keyboard or the touch screen with messages on the screen or the audio system. This allows the student to work at his own pace, for as long as he wants. Then, the student's performance and other data is stored from one use of the lesson to the next.

1.3 Lessons in PLATO IV

Lessons are written in PLATO using TUTOR, a computer language that allows communication with the computer. The lessons can be written in the system using any terminal, and students can use these lessons from any terminal. Several students can take the same lesson at the same time, and the system can treat them as individuals, or as a group interacting with each other.

2. AUTOMATION OF C.S. 100-LEVEL COURSES

The project to automate all C.S. 100-level courses has three basic components: a computer system with capabilities of interaction (PLATO IV), a library of lessons, and a conversational information-retrieval system, called the GUIDE.

2.1 Library of Lessons

Since introductory courses are offered to students with many different backgrounds, a large number of lessons should be available in the library. There will be introductory lessons which will be self-contained, that is, they will need no prerequisite. Also, there will be sequences of lessons for some of the common programming languages, and lessons with examples and applications for the different fields in which computers are used.

From these lessons an instructor can choose a subset for his course. This subset of lessons will be required of all the students in the course, although they may take some other lessons of particular interest to them from among those in the library.

2.2 The GUIDE

There will be one lesson called GUIDE containing a catalog of the lessons in the library, with the contents of the lessons and information about relationships among lessons. This lesson will also keep records of the lessons a student has taken and of his performance, and a schedule for the lessons required of the student.

The student would go to GUIDE to get suggestions on which lesson he could or should take next, according to the schedule for his course, his performance on those required lessons, and his interests. In this way students

from different departments will be able to take different lessons among those in the library depending on their particular interests or backgrounds, besides those lessons required in the course.

2.3 Lessons on PL/I

The library of lessons will contain sequences of lessons on PL/I as well as on other programming languages. For a programming language, a sequence of lessons is needed because of the difficulty of making the lessons self-contained. Thus, the student should follow a certain order when taking the lessons of a sequence.

2.4 PL/I Sequence

Some of the lessons in the PL/I sequence will be:

pl1data: an introduction to the four basic data types used in PL/I--fixed point numbers, floating point numbers, character strings, and bit strings;

pl1ops: introductory lesson on operations with the four data types;

pl1if: covers the PL/I statements IF THEN and IF THEN ELSE;

pl1do: lesson on DO loops in PL/I--DO WHILE loops, indexed DO loops, and nesting of DO loops;

pl1array: theory and applications of arrays in PL/I.

There will be some other lessons on PL/I. One of them is a compiler that will allow the student to write short programs and get appropriate error messages if his program is wrong. This lesson will even "execute" the program for the student.

2.5 Intended Audience of the PL/I Sequence

Although the lessons in the PL/I sequence are designed for students taking introductory C.S. courses, they will have material useful for students

in other departments or for students with more computer experience.

It is intended that each lesson covers the subject completely, so that even experienced programmers will find the lesson useful.

2.6 Prerequisites

No previous computer background is required to take the lessons in the PL/I sequence, but the student is expected to be enrolled in an introductory C.S. course and to have already taken a few weeks of it. The lessons are written in such a way that a student can learn from them after a few weeks of his first C.S. course.

3. ARRAYS IN PL/I

Since some other authors were already writing lessons for the PL/I sequence, I selected arrays in PL/I as the topic of my lesson from among those lessons that still needed to be written for the sequence.

3.1 Purpose of the Lesson

I did not intend to prove the usefulness of arrays, but to teach the student how to use arrays in PL/I properly and, hopefully, in an efficient manner.

My goal was to write a lesson that would be:

--simple and clear because in this way the student learns more and learns faster;

--objective so the student would understand the essence of the topic;

--interesting enough to hold the student's attention so he would retain more of the lesson;

--relevant so he wouldn't get discouraged by too much extra material and wouldn't distract his attention from the main idea;

--complete so that the student could find everything about the subject in the lesson; and

--easy to use so he can find what he is looking for easily and avoid confusion on what to do next or how to do it.

3.2 Development of the Lesson

With these goals in mind, I first familiarized myself with the PLATO IV system and with its language, TUTOR.

Next, I wrote the table of contents:

THEORY

1. Array basics
2. Declaration of arrays
 - A. Declaration of linear arrays
 - B. Declaration of multidimensional arrays
 - C. Array declaration drill
 - D. The INITIAL attribute
 - E. Drill on the INITIAL attribute
3. Operations with entire arrays
4. Operations with parts of an array
5. Input/Output of arrays

APPLICATIONS

6. Matrix multiplication
7. Connectivity matrix (graphs)
8. Distance matrix (graphs)

This would help me to keep in mind the different topics of the lesson, and to be systematic and uniform while developing it. I divided the lesson in two parts: theory and applications.

The part on theory was based on references [2] through [8].

For the applications, I wrote programs for some interesting topics suggested by Professor Nievergelt. I ran these programs in the IBM System/360, along with some others containing statements from the theory section of the lesson, to make sure that all the PL/I statements and programs in the lesson worked as intended.

Before programming the lesson in PLATO, I wrote it on paper, keeping in mind the most important features of the PLATO system to insure the best use

of them. Next, while writing the lesson in PLATO, I took advantage of its display facilities to make the lesson interesting and easy to use.

In some respects, the lesson is like a book:

--it has a table of contents that can be reached from anywhere in the lesson;

--it is possible to go from the table of contents to the first page of any topic; and

--the following as well as the preceding pages are accessible any time by pressing a single key.

In other respects, the lesson is much better than a book. Part of one page can be displayed to help the student concentrate on one new idea at a time, followed then by some examples to help him understand it completely. Then another idea is presented, and so on. Additional text or figures can be added to the same page, or parts of the page can be erased and new things displayed. I used this feature frequently because it helps the student visualize things that words alone cannot explain.

Because of restrictions in memory space, I decided to finish with the theory and the applications first, and then add some drills for the student.

3.3 Trial with Students

On May 16, 1973, the lesson was tested with students. Although the trial was small, only ten students, it was enough to give some idea of how good the lesson was and what changes could be done for improvement.

The students were asked to fill in a lesson evaluation form for ratings, comments, and suggestions. On the average, these students had taken eight other lessons in PLATO. The average time spent in completing the lesson was 42 minutes. Seventy-five percent of the students had thorough previous knowledge of the lesson material. The ratings for the lesson were as follows:

		Average rating:
Examples:	too many = 4, too few = 1	2.7
	very clear = 4, unclear = 1	2.9
	relevant = 4, irrelevant = 1	3.1
Exercises:	too many = 4, too few = 1	1.4
	very helpful = 4, worthless = 1	2.5
	very clear = 4, unclear = 1	2.7
Text material:	too much = 4, too little = 1	3.0
	very clear = 4, unclear = 1	3.1
Displays:	excellent = 4, poor = 1	3.5
Total lesson:	learned a lot = 4, learned nothing = 1	2.8
	easy to use = 4, hard to use = 1	3.4
	too much material = 4, too little = 1	3.2
	much better than a book = 4, much worse = 1	3.2
	much better than a lecture = 4, much worse = 1	3.8

As expected, the students complained about too little student input or student interaction with the lesson. Two general comments were that there was too much material to be covered in a single lesson, and that the two applications on graphs were way over the heads of 100-level students.

3.4 Future of the Lesson

I have made several changes and additions since the trial. There is virtually no memory space left to add drills and exercises for the student. Because the two applications on graphs seem to belong to a different lesson, they might be removed from this lesson and put in another. This would provide more space for drills, which would mean more interaction for the student. It would also allow the improvement of some sections of the lesson that are weak and need some additions.

The lesson is not perfect yet, but it is good enough now to be used by the students with profit. Being easier to modify than a book, changes can and will be made to the lesson to integrate it with the other lessons in the PL/I sequence.

LIST OF REFERENCES

[1] D. Alpert, and D. L. Bitzer, "Advances in Computer Based Education," Science, No. 167, pp. 1582-1590; 1970.

[2] Frank Bates, and Mary L. Douglas, Programming Language/ONE, Second Edition, Prentice-Hall, Inc.; 1970.

[3] Gabriel F. Groner, PL/I Programming in Technological Applications, Wiley-Interscience; 1971.

[4] International Business Machines Corporation, "System/360 Operating System, PL/I (F) Language Reference Manual," New York; June 1970.

[5] Michael Kennedy, and Martin B. Solomon, Eight Statement PL/C (PL/ZERO) plus PL/ONE, Prentice-Hall, Inc.; 1972.

[6] Erich J. Neuhold, and Harold W. Lawson, Jr., The PL/I Machine: An Introduction to Programming, Addison-Wesley; 1971.

[7] David Stoutemeyer, PL/I Programming for Engineering and Science, Prentice-Hall, Inc.; 1971.

[8] Andrew Vazsonyi, Problem Solving by Digital Computers with PL/I Programming, Prentice-Hall, Inc.; 1970.

APPENDIX

TUTOR PROGRAM "ARRAYS IN PL/I"

```
                                                                                                                    START
  2  EDIT     N40
  3  DEFINE   HEREX=N1,HEREY=N2,ROWS=N3
  4           COLS=N4,SISX=N5,SISY=N6
  5           COUNT=N7,STRING=N8,STRLEN=N12
  6           NAME=N13,LNAME=N15,AQUI=N16
  7           PTR=N17,FLAG=N18,R1=N19
  8           R2=N20,FLEM=N21,DIM=N22,FLAG1=N23
  9           LP=N24,RP=N25,CM=N26,CL=N27,SC=N28
 10  CALC     SC*17+CM*4,RP*1+LP*1+CL*40
 11  CHAR     1
 12           0140000,0170000,0036000,0007400
 13           0001700,0000360,0000074,0000017
 14  CHAR     2
 15           0000017,0000074,0000360,0001700
 16           0007400,0036000,0170000,0140000
 17  CHAR     3
 18           0177777,0177777,0177777,0177777
 19           0177777,0177777,0177777,0177777
 20  CHAR     8
 21           0140003,0160007,0170017,0154033
 22           0146063,0143143,0141703,0140603
 23  CHAR     9
 24           0140003,0140003,0160007,0170017
 25           0000000,0000000,0000000,0000000
 26  CHAR     37
 27           0000700,0001240,0002220,0003760
 28           0002220,0001240,0000700,0000000
 29  CHAR     52
 30           0000700,0001040,0002520,0002220
 31           0002520,0001040,0000700,0000000

                                                                                                                    START
 32  UNIT     START
 33  AT       202
 34  WRITE    CS100
 35  FIGURE   101,201,208,108,101
 36  SIZE     5
 37  AT       1217
 38  WRITE    *A*R*R*A*Y*S
 39  AT       2015
 40  WRITE    *I*N *P*L/1
 41  SIZE     0
 42  AT       3240
 43  WRITE    *BY *LUTS *H *JAUREGUI
 44  PAUSE
 45  ERASE
 46  AT       101
 47  WRITE    *A*R*R*A*Y*S *I*N *P*L/1
 48  AT       610
 49  WRITE    *NEXT IS THE TABLE OF CONTENTS.
 50           *IT IS SUGGESTED THAT YOU GO THROUGH THE LESSON
 51           IN NUMERICAL ORDER ALTHOUGH YOU CAN STUDY THE
 52           TOPICS IN ANY ORDER YOU WANT.
```

```
53  AT      1310
54  WRITE   *YOU CAN REACH THE TABLE OF CONTENTS FROM
55          ANYWHERE IN THE LESSON FOR REVIEW OR FOR A
56          DIFFERENT TOPIC BY PRESSING THE *T*E*R*M KEY AND
57          TYPING *7CONTENTS*7 OR *7INDEX*7.
58  AT      2010
59  WRITE   *THE *N*E*X*T KEY WILL USUALLY BRING YOU
60          THROUGH A TOPIC UNLESS OTHERWISE SPECIFIED.
61          *THE *B*A*C*K KEY WILL BRING YOU BACK TO THE
62          PREVIOUS FRAME.
```

INDEX=HAS.

CONTENTS

```
64  UNIT    CONTENTS
65  TERM    CONTENTS
66  TERM    INDEX
67  BASE
68  ERASE
69  AT      101
70  WRITE   *A*R*R*A*Y*S *I*N *P*L/1
71  AT      515
72  SIZE    2
73  WRITE   *C*O*N*T*E*N*T*S
74  SIZE    0
75  AT      805
76  WRITE   *T*H*E*O*R*Y
77  AT      1008
78  WRITE   1  *ARRAY BASICS
79          2  *DECLARATION OF ARRAYS
80          3  *OPERATION WITH ENTIRE ARRAYS
81          4  *OPERATION WITH PARTS OF AN ARRAY
82          5  *INPUT/*OUTPUT OF ARRAYS
83  AT      1605
84  WRITE   *A*P*P*L*I*C*A*T*I*O*N*S
85  AT      1808
86  WRITE   6  *MATRIX MULTIPLICATION
87          7  *CONNECTIVITY  MATRIX (GRAPHS)
88          8  *DISTANCE MATRIX (GRAPHS)
89  AT      2205
90  WRITE   *TYPE THE NUMBER OF THE TOPIC YOU WANT TO STUDY (TYPE
91          0 IF YOU WANT TO QUIT). *YOUR LAST TOPIC WAS NUMBER  .$$
92  SHOW    N29.1
93  AT      2805
94  WRITE   *AT THE END OF EACH TOPIC YOU WILL RETURN TO THIS PAGE.
95          *YOU CAN RETURN HERE ANYTIME BY PRESSING THE *T*E*R*M
96          KEY AND TYPING *7CONTENTS*7 OR *7INDEX*7.
97  ARROW   2505
98  LONG    1
99  MATCH   N29.0,1,2,3,4,5,6,7,8
100 JUDGE   N29.IGNORE,OK
101 JUMP    N29,X,ENDLESS,BASICS,DECLARAT,ENTIRE,PARTS,ARRAYI/O
102         MATRMULT,CONNMATR,DISTMATR
```

```
103 UNIT    BASICS
104 BACK    CONTENTS
105 ERASE
106 AT      101
107 WRITE   *A*R*R*A*Y *H*A*S*I*C*S
108 AT      610
109 WRITE   *SOMETIMES DURING COMPUTATION WE ARE DEALING
110         WITH SEVERAL DATA ITEMS THAT ARE SIMILAR.
111         *INSTEAD OF GIVING EACH ONE A VARIABLE NAME,
112         WE CAN GROUP THEM INTO LISTS OR TABLES AND
113         GIVE THEM A SINGLE VARIABLE NAME.
114
115         *LISTS, TABLES, AND OTHER SIMILAR COLLECTIONS
116         OF DATA ITEMS ARE CALLED *A*R*R*A*Y*S IN *P*L/1.
117
118         *AN *ARRAY IS AN ORDERED COLLECTION OF ELEMENTS
119         WITH IDENTICAL DATA ATTRIBUTES GROUPED IN A LIST,
120         A TABLE, OR ANY SIMILAR AGGREGATE.
121
122         *EACH OF THE ELEMENTS OF THE ARRAY IS CALLED
123         A *S*U*B*S*C*R*I*P*T*E*D *V*A*R*I*A*B*L*E.
```



```
124 UNIT    AA
125 BACK    BASICS
126 ERASE
127 AT      505
128 WRITE   *LET*'S SUPPOSE WE HAVE THE FOLLOWING LIST OF CARS*:
129         *FORD, *DODGE, *FIAT, *MERCEDES, *VOLKSWAGEN, *ROLLS *ROYCE,
130         AND *CADILLAC.
131         *WE CAN GROUP THEM UNDER NAME *C*A*R AND REFER TO EACH
132         INDIVIDUAL BRAND BY THE NAME *C*A*R WITH A SUBSCRIPT*:
133 AT      1125
134 WRITE       *C*A*R
135         *F*O*R*D
136         *D*O*D*G*E
137         *F*I*A*T
138         *M*E*R*C*E*D*E*S
139         *V*O*L*K*S*W*A*G*E*N
140         *R*O*L*L*S *R*O*Y*C*E
141         *C*A*D*I*L*L*A*C
142 CALC    HEREY=128+HEREX=181+SISY=4+ROWS=5+SISY=COLS=1
143 JOIN    DRAW
144 PACK    STRLEN,STRING,*C*A*R
145 CALC    HERFY=185+HERFX=135
146 JOIN    FILL
147 AT      1221
148 WRITE   v1
149         v2
150         v3
151         v4
152         v5
153         v6
154         v7
155 INHIBIT ERASE
```

```
                                                                                                                    AAA
156  UNIT    AAA
157  BACK    BASICS
158  AT      1221

------------------------------------------------------------------------------------------------------------------ BASICS
160  MODE    REWRITE
161  WRITE   (1)
162          (2)
163          (3)
164          (4)
165          (5)
166          (6)
167          (7)
168  MODE    WRITE
169  AT      2005
170  WRITE   *IN *P*L/1 THE SUBSCRIPT APPEARS IN PARENTHESES AFTER
171          THE NAME OF THE ARRAY (*S*U*B*S*C*R*I*P*T*E*D *V*A*R*I*A*B*L*E).
172
173          *C*A*R(1) IS THE FIRST ELEMENT OF THE ARRAY AND CONTAINS *F*O*R*D.
174          *C*A*R(3) CONTAINS *F*I*A*T.
175          *EACH OF THE ELEMENTS OF AN ARRAY ACTS AS A VARIABLE.
176          *FOR EXAMPLE*:
177                       *C*A*R(3)=*7*P*I*N*T*O*7;
178          WILL CHANGE THE THIRD ELEMENT TO *P*I*N*T*O.
179          (*PRESS *N*E*X*T TO SEE)
180  INHIBIT ERASE

                                                                                                                    AAB
181  UNIT    AAB
182  BACK    BASICS
183  MODE    ERASE
184  AT      1425
185  WRITE   *F*I*A*T
186  MODE    WRITE
187  AT      1425
188  WRITE   *P*I*N*T*O

                                                                                                                    AB
189  UNIT    AB
190  BACK    AA
191  AT      405
192  WRITE   *IF WE HAVE A TABLE CONTAINING THE HIGH AND LOW
193          TEMPERATURES OF SOME WEEK, WE CAN STORE IT IN A
194          TWO-DIMENSIONAL ARRAY CALLED *T*E*M*P*;
195  CALC    HEREY*263*HEREX*105
196  PACK    STRLEN,STRING,*T*E*M*P( , )
197  JOIN    FILL
198  CALC    HEREX*178*SISX*4*COLS*2
199  JOIN    DRAW
200  CALC    HEREX*289
201  JOIN    FILL
202  AT      810
203  WRITE            *T*E*M*P
```

```
        LESSON PL1ARRAY      AT     17.21.24.    ON     06/13/73

204             1 1       50     37       1 2
205             2 1       52     35       2 2
206             3 1       48     31       3 2
207             4 1       49     33       4 2
208             5 1       49     30       5 2
209             6 1       58     37       6 2
210             7 1       64     45       7 2
211     INITIAT ERASE

                                                                                ABA
212     UNIT    ABA
213     BACK    AA
214     AT      1705
215     WRITE   +EVERY ROW REPRESENTS ONE DAY OF THE WEEK (ROW 1 FOR
216             +SUNDAY, ROW 2 FOR +MONDAY, ETC.), COLUMN 1 CONTAINS
217             THE HIGH TEMPERATURES, AND COLUMN 2 CONTAINS THE
218             LOW TEMPERATURES.
219
220             +ARRAY +T+E+M+P HAS 14 ELEMENTS IN TWO DIMENSIONS
221             (7 ROWS AND 2 COLUMNS).
222
223             +WE NEED ONE SUBSCRIPT FOR EACH DIMENSION, AND
224             THEY MUST BE SEPARATED BY COMMAS FROM EACH OTHER.
225
226             +T+E+M+P(4,1) IS THE ELEMENT IN ROW 4 (+WEDNESDAY) AND
227             COLUMN 1 (HIGH TEMPERATURE), AND CONTAINS THE HIGH
228             TEMPERATURE OF +WEDNESDAY (49¬0+F).

                                                                                AC
229     UNIT    AC
230     BACK    AB
231     AT      1010
232     WRITE   +IF WE WANT TO STORE THE HIGH AND LOW
233             TEMPERATURES OF EVERY DAY OF THE YEAR,
234             WE WOULD NEED A THREE-DIMENSIONAL ARRAY
235             WITH 12 ROWS (12 MONTHS), 31 COLUMNS
236             (31 DAYS), AND 2 PLANES (HIGH AND LOW).
237     AT      1710
238     WRITE   +T+E+M+P(5,27,1) WOULD CONTAIN THE HIGH (1)
239             TEMPERATURE OF THE 27TH DAY OF THE 5TH
240             MONTH,I.E. THE HIGH TEMPERATURE OF THE
241             27TH OF +MAY.

                                                                                AD
242     UNIT    AD
243     BACK    AC
244     WRITE   +A+R+R+A+Y +B+A+S+I+C+S
245     AT      912
246     WRITE   +A SUBSCRIPTED NAME IS USED TO IDENTIFY
247             AND REFFRENCE A SINGLE ELEMENT OF AN
248             ARRAY. +THE FORM OF THE SUBSCRIPTED NAME IS+:
249

-------------------------------------------------------------------------------- ?CL-1 INEAR
251               ANAME<<<<<<+6+6+6+6+6 ( SUBS1<<<<<+6+6+6+6+6,SUBS<<<<+6+6+6+62<+6,...)
```

```
252  AT       1512
253  WRITE    ANAME<<<<<*6*6*6*6*6 IS THE NAME OF THE ARRAY AND
254           SUB1<<<<*6*6*6*6,SUB2<<<<*6*6*6*6*,... ARE THE SUBSCRIPTS.
255
256           *WE MUST HAVE ONE SUBSCRIPT FOR EVERY
257           DIMENSION OF THE ARRAY.
258  NEAT     ENDTOPIC

                                                                                DECLARAT
259  UNIT     DECLARAT
260  BACK     CONTENTS
261  ERASE
262  AT       101
263  WRITE    *A*R*R*A*Y*S *I*N *P*L/1
264  AT       717
265  WRITE    *D*E*C*L*A*R*A*T*I*O*N *O*F *A*R*R*A*Y*S
266  AT       1010
267  WRITE    *A       *DECLARATION OF LINEAR ARRAYS
268           *B       *DECLARATION OF MULTIDIMENSIONAL ARRAYS
269           *C       *DRILL ON ARRAY DECLARATION
270           *D       *THE *I*N*I*T*I*A*L ATTRIBUTE
271           *E       *DRILL ON THE *I*N*I*T*I*A*L ATTRIBUTE
272           *F       *THE *C*O*N*T*E*N*T*S PAGE
273  AT       1710
274  WRITE    *TYPE THE LETTER OF THE TOPIC YOU WANT
275  ARROW    1810
276  LONG     1
277  MATCH    N30,*A,*B,*C,*D,*E,*F
278  JUDGE    N30,*IGNORE,OK
279  JUMP     N30-1,LINEAR,MULTIDIM,DRILL1,INITIAL,DRILL2,CONTENTS

                                                                                LINEAR
280  UNIT     LINEAR
281  BACK     DECLARAT
282  ERASE
283  AT       101
284  WRITE    *L*I*N*E*A*R *A*R*R*A*Y *D*E*C*L*A*R*A*T*I*O*N
285  AT       310
286  WRITE    *LINEAR OR ONE-DIMENSIONAL ARRAYS ARE DECLARED
287           IN *P*L/1 AS FOLLOWS*!
288  AT       608
289  WRITE    *D*E*C*L*A*R*E ANAME<<<<<*6*6*6*6*6 (BOUND<<<<<*6*6*6*6*6) A<*6TTR<<<*6*6*6!
290           OR  *D*C*L ANAME<<<<<*6*6*6*6*6 (BOUND<<<<<*6*6*6*6*6) AT<<<*6*6TR<<*6*6!
291  AT       912
292  WRITE    ANAME<<<<<*6*6*6*6*6 IS ANY LEGAL *P*L/1 VARIABLE NAME.
293           BOUND<<<<<*6*6*6*6*6 IS A POSITIVE DECIMAL INTEGER THAT
294                 SPECIFIES THE UPPER BOUNDARY FOR THE
295                 SUBSCRIPT.
296           ATTR<<<<*6*6*6*6  ARE  *P*L/1 ATTRIBUTES THAT SPECIFY
297                 THE DATA CHARACTERISTICS FOR EVERY ELEMENT
298                 OF THE ARRAY. *IF OMMITED, *P*L/1 ASSIGNS
299                 ATTRIBUTES BY DEFAULT, ACCORDING TO THE
300                 FIRST LETTER OF ANAME<<<<<*6*6*6*6*6.
301  INHERIT ERASE
```

```
302  UNIT    B
303  BACK    DECLARAT
304  AT      1905
305  WRITE   ROUND$$$$$*6*6*6*6*6 CAN BE A VARIABLE IF THE DECLARATION IS
306          INSIDE A PROCEDURE OR BEGIN BLOCK AND THE VARIABLE
307          HAS BEEN GIVEN AN INTEGER VALUE OUTSIDE THE BLOCK*!
308
309                                  *M*P*R*!*P*R*O*C *O*P*T*I*O*N*S(*M*A*I*N)!
310          ....                          ....
311          *X=27!                         *M=9!
312          *H*E*G*I*N!                        *C*A*L*L *S*R*R!
313            *D*C*L *A(*X)!                   ...
314          ...                        *S*B*R*!*P*R*O*C!
315          *E*N*D!                           *D*C*L *A*R*R(*M,*M)!
316          ...                           ...
317                                        *E*N*D *S*R*R!
318                                    *E*N*D *M*P*R!

                                                                                                        BA
319  UNIT    BA
320  BACK    LINEAR
321  WRITE   *D*C*L ANAME (ROUND) ATTR!
322  AT      B0G
323  WRITE   *D*C*L *A(5) *E*I*X*E*D *B*I*N(31)!          1
324                                      2                                                                  19
 ------------------------------------------------------------------------------------------------------- LINEAR
326          DEFINES LINEAR ARRAY *A          3
327          AS HAVING 5 ELEMENTS WITH        4
328          ATTRIBUTES *E*I*X*E*D *B*I*N*A*R*Y(31).    5
329  CALC    HEREY*BB*HEREX*2B5*SISX*6*ROWS*4*COLS*SISY*1
330  BACK    STOLEN,STRING,*A( )
331  JOIN    FILL
332  CALC    HEREX*336
333  JOIN    DRAW
334  AT      748
335  WRITE   *A
336  INHIBIT ERASE

                                                                                                        BAA
337  UNIT    BAA
338  BACK    LINEAR
339  AT      1B24
340  WRITE   1     *D*C*L *B(4) *L*A*B*E*L!
341          2
342          3     DECLARES ARRAY *B AS LINEAR WITH
343          4     FOUR ELEMENTS THAT WILL BE LABELS
344                FOR THE PROGRAM.
345  CALC    HEREY*56*HEREX*164*ROWS*4
346  BACK    STOLEN,STRING,*B( )
347  JOIN    FILL
348  CALC    HEREX*72
349  JOIN    DRAW
350  AT      1715
```

```
351  WRITE   *B

                                                                        BB
352  UNIT    BB
353  BACK    BA
354  WRITE   *D*C*L ANAME (BOUND) ATTR!
355  AT      B15
356  WRITE   *WE CAN STORE THE NAMES OF THE
357          12 STUDENTS IN SOME CLASS IN
358          ARRAY *S*T*U*D*E*N*T DECLARED AS FOLLOWS*!
359
360          *D*C*L *S*T*U*D*E*N*T(12) *C*H*A*R(20)!
361  AT      1518
362  WRITE               *S*T*U*D*E*N*T
363           1
364           2
365           3
366           4
367           5
368           6
369           7
370           8
371           9
372          10
373          11
374          12
375  CALC    HEREY*184*HEREX*52*SISX*8*ROWS*12
376  PACK    STRLEN,STRING,*S*T*U*D*E*N*T(  )
377  JOIN    FILL
378  CALC    HEREX*160
379  JOIN    DRAW

                                                                        BC
380  UNIT    BC
381  BACK    BB
382  WRITE   *L*I*N*E*A*R *A*R*R*A*Y *D*E*C*L*A*R*A*T*I*O*N
383  AT      910
384  WRITE   *SO FAR WE HAVE USED 1 AS THE SUBSCRIPT OF
385          THE FIRST ELEMENT OF AN ARRAY. *SOMETIMES IT
386          IS CONVENIENT TO HAVE THIS *.LOWER BOUNDARY*.
387          DIFFERENT THAN 1. *THEN THE LINEAR ARRAY IS
388          DECLARED AS FOLLOWS*!
389
390            *D*C*L ANAME<<<<<<*6*6*6*6*6 ( LBOUND<<<<<<<*6*6*6*6*6*6*!UBO<<<*6*6*6
391  WRITE   UND<<<*6*6*6 ) ATTR<<<<*6*6*6*6*!
392  AT      1910
393  WRITE   LBOUND<<<<<<<*6*6*6*6*6*6 SPECIFIES THE LOWER BOUNDARY AND
394          UBOUND<<<<<<<*6*6*6*6*6*6 THE UPPER BOUNDARY FOR THE SUBSCRIPT.
395          *A COLON '(*!) MUST SEPARATE LBOUND<<<<<<<*6*6*6*6*6*6 AND UB<<*6*6
396  WRITE   OUND<<<<*6*6*6*6.
397
398          LBOUND<<<<<<<*6*6*6*6*6*6 HAS TO BE LESS THAN OR EQUAL
399          TO UBOUND<<<<<<<*6*6*6*6*6*6.

                                                                        BD
```

```
400  UNIT    BD
401  BACK    BC
402  ERASE
403  AT      101
404  WRITE   *D*C*L ANAME(BOUND)ATTR;        *D*C*L ANAME(LBOUND*:UBOUND)ATTR;
405  AT      1012
406  WRITE   *THUS, THE FOLLOWING DECLARATIONS DEFINE
407          TWO ARRAYS OF FOUR ELEMENTS EACH, BUT EACH
408          ELEMENT IS REFERED TO IN A DIFFERENT WAY*;
409
410              *D*C*L *A(4);          *D*C*L *A(0*:3);
411                    *A                     *A
412              1                    0
413              2                    1
414              3                    2
415              4                    3
416  CALC    HEREY*152*HEREX*98,SISX*2*ROWS*4
417  PACK    STRLEN,STRING**A( )
418  JOIN    FILL

------------------------------------------------------------------------------------------------------------------------ LIN_MULTID
420  CALC    HEREX*144
421  JOIN    DRAW
422  CALC    HEREX*256
423  JOIN    FILL
424  CALC    HEREX*296
425  JOIN    DRAW

                                                                                                                        BE
426  UNIT    BE
427  BACK    BD
428  WRITE   *D*C*L ANAME (LBOUND*:UBOUND) ATTR;
429  AT      505
430  WRITE   *WE CAN STORE THE
431          COEFFICIENTS OF AN
432          EQUATION OF THE FORM*;
433
434          AV0*AV1X*AV2X^2*AV3X^3=0
435
436          IN AN ARRAY DECLARED
437          AS FOLLOWS*;
438
439          *D*C*L *A(0*:3);
440                *A
441            0
442            1
443            2
444            3
445  CALC    HEREY*224*HEREX*28,ROWS*4
446  PACK    STRLEN,STRING**A( )
447  JOIN    FILL
448  CALC    HEREX*64
449  JOIN    DRAW
450  INMIDIT ERASE
```

```
                                                                                                    BEA
451  UNIT    BEA
452  BACK    BD
453  AT      530
454  WRITE   *OR STORE THE GROSS PROFIT OF
455          A COMPANY FROM 1965 THROUGH
456          1973 IN ARRAY *G*R*O*S*S*6*P*R*O*F*I*T*!
457
458          *D*C*L *G*R*O*S*S*6*P*R*O*F*I*T(1965*:1973)!
459  AT      1143
460  WRITE     *G*R*O*S*S*6*P*R*O*F*I*T
461          1965
462          1966
463          1967
464          1968
465          1969
466          1970
467          1971
468          1972
469          1973
470  CALC    HEREY*88*HEREX*223*ROWS*9
471  PACK    STRLEN,STRING**G*R*O*S*S*6*P*R*O*F*I*T(     )
472  JOIN    FILL
473  CALC    HEREX*376
474  JOIN    DRAW
475  AT      2305
476  WRITE   *IN THIS WAY WE CAN REFER TO AN ELEMENT IN A MORE
477          MEANINGFUL WAY*!
478          *A(3) IS THE COEFFICIENT OF X^3.
479          *G*R*O*S*S*6*P*R*O*F*I*T(1969) IS THE COMPANY*'S GROSS PROFIT
480          IN 1969.
481  NEXT    ENDTOPIC
                                                                                                    HCA
482  UNIT    HCA
483  BACK    DECLARAT
484  AT      2505
485  WRITE   ANAME<<<<<<*6*6*6*6*6 IS ANY LEGAL *P*L/1 VARIABLE NAME.
486          ATTR<<<<<*6*6*6*6  ARE  *P*L/1 ATTRIBUTES THAT SPECIFY
487                THE DATA CHARACTERISTICS FOR EVERY ELEMENT
488                OF THE ARRAY. *IF OMMITED. *P*L/1 ASSIGNS
489                ATTRIBUTES BY DEFAULT, ACCORDING TO THE
490                FIRST LETTER OF ANAME<<<<<<*6*6*6*6*6.
491  INHERIT ERASE
492  NEXT    C
                                                                                                    HCH
494  UNIT    HCB
494  JOIN    LINEAR
495  JOIN    H
496  AT      157
497  WRITE   (*REVIEW)
498  NEXT    HCBA
                                                                                                    HCHA
```

```
499  UNIT    HCRA
500  ERASE
501  JOIN    MULTIDIM
502  JUMP    C

                                                                                                    MULTIDIM
503  UNIT    MULTIDIM
504  BACK    DECLARAT
505  HELP    HCA
506  AT      101
507  WRITE   *M*U*L*T*I*D*I*M*E*N*S*I*O*N*A*L *A*R*R*A*Y *D*E*C*L*A*R*A*T*I*O*N
508  AT      305
509  WRITE   *ARRAYS WITH MORE THAN ONE DIMENSION ARE
510          DECLARED AS FOLLOWS*!
511
512          *D*C*L ANAME$$$$$*6*6*6*6*6 (BOUND1$$$$$$*6*6*6*6*6,BOU$$<*6*6*6ND$$*6*62<*6,...) ATTR<<<<*6*6*6*6!
513
514          ANAME$$$$$*6*6*6*6*6 AND ATTR$$$$*6*6*6*6 ARE THE SAME AS FOR LINEAR ARRAYS.
515          (*PRESS *H*E*L*P FOR REVIEW)
516
517          BOUND1$$$$$$*6*6*6*6*6*6 IS THE UPPER BOUNDARY FOR SUBSCRIPT 1.
518          BOUND2$$$$$$*6*6*6*6*6*6 IS THE UPPER BOUNDARY FOR SUBSCRIPT 2.
519          ...
520
521          *ONE UPPER BOUNDARY PER DIMENSION IS REQUIRED.
522          *ALL THE LOWER BOUNDARIES ARE 1.

-------------------------------------------------------------------------------------------------------- MULTIDIMEN
524          *ONE COMMA IS USED TO SEPARATE ONE UPPER BOUNDARY
525          FROM THE NEXT.
526  INHIBIT ERASE

                                                                                                    C
527  UNIT    C
528  BACK    DECLARAT
529  HELP    HCR
530  AT      2005
531  WRITE   BOUND1$$$$$$*6*6*6*6*6*6,BOUND2$$$$$$*6*6*6*6*6*6,... MUST BE POSITIVE DECIMAL
532          INTEGERS, UNLESS THE DECLARATION IS INSIDE A
533          BLOCK AS SHOWN IN *DECLARATION OF *LINEAR *ARRAYS.
534          (*PRESS *H*E*L*P TO REVIEW)

                                                                                                    CA
535  UNIT    CA
536  BACK    MULTIDIM
537  ERASE
538  AT      101
539  WRITE   *D*C*L ANAME (BOUND1,BOUND2,...) ATTR!
540  AT      510
541  WRITE   *D*C*L *R*E*C*T*A*N(12,3)!
542
543          *R*E*C*T*A*N IS A TWO-DIMENSIONAL ARRAY WITH
544          12 ROWS AND 3 COLUMNS, I.E. 36 ELEMENTS.
545
546                          *R*E*C*T*A*N
```

```
        LESSON PL1ARRAY      AT     17.21.26.    ON     06/13/73

547  CALC    HEREY=190+HEREX=132+ROWS=6+SISX=3+COLS=2+SISY=1
548  JOIN    DRAW

                                                                                                                    CB

549  UNIT    CB
551  BACK    CA
551  WRITE   *MULTIDIMENSIONAL *ARRAY *DECLARATION
552  AT      808
553  WRITE        *D*C*L *C*U*B*E(10,10,10);
554
555          *C*U*B*E IS A THREE-DIMENSIONAL ARRAY WITH 10
556          ROWS, 10 COLUMNS, AND 10 PLANES, I.E. 1000
557          ELEMENTS (10*10*10=1000).
558  INHIBIT ERASE

                                                                                                                   CBA

559  UNIT    CBA
560  BACK    CA
561  AT      1608
562  WRITE   *ARRAYS IN *P*L/1 MAY HAVE ANY NUMBER OF
563          DIMENSIONS. BUT IT IS EASY TO RUN OUT OF
564          MEMORY SPACE IN THAT WAY. *CONSIDER FOR EXAMPLE:*;
565
566                  *D*C*L *B*I*G(5,8,6,7,9,4) *C*H*A*R(9);
567
568          *ARRAY *B*I*G CONTAINS 5*8*6*7*9*4=68,480 ELEMENTS AND
569          EACH ELEMENT NEEDS 9 BYTES OF MEMORY (*C*H*A*R(9)).
570          *ARRAY *B*I*G NEEDS THEN 616,320 BYTES OF MEMORY,
571          WHICH IS WASTEFUL, IF POSSIBLE AT ALL.

                                                                                                                    CC

572  UNIT    CC
573  BACK    CB
574  WRITE   *MULTIDIMENSIONAL *ARRAY *DECLARATION
575  AT      505
576  WRITE   *WHEN WE WANT THE LOWER BOUNDARIES FOR THE SUBSCRIPTS
577          TO BE OTHER THAN 1, THE *D*E*C*L*A*R*E STATEMENT IS*;
578
579          *D*C*L ANAME(<<<<<*6*6*6*6*6(LBOUND1:<<<<<<<*6*6*6*6*6*6*6*,UBOUND1:<<<<<<<*6*6*6*6*6*6*6,LBOUN<<<<<*6*6*6*6*6*
580  WRITE   D2<<<*6*6*,UBOUND2:<<<<<<<*6*6*6*6*6*6*6,...)ATTR(<<<<<*6*6*6*6);
581
582            LBOUND1:<<<<<<<*6*6*6*6*6*6*6 IS THE LOWER BOUNDARY FOR SUBSCRIPT 1.
583            UBOUND1:<<<<<<<*6*6*6*6*6*6*6 IS THE UPPER BOUNDARY FOR SUBSCRIPT 1.
584            LBOUND2:<<<<<<<*6*6*6*6*6*6*6 IS THE LOWER BOUNDARY FOR SUBSCRIPT 2.
585            UBOUND2:<<<<<<<*6*6*6*6*6*6*6 IS THE UPPER BOUNDARY FOR SUBSCRIPT 2.
586            ...
587
588          *A COLON SEPARATES THE LOWER BOUNDARY FROM
589          THE UPPER FOR ANY SUBSCRIPT, AND A COMMA
590          SEPARATES THE BOUNDARIES FOR ONE SUBSCRIPT
591          FROM THE BOUNDARIES FOR THE NEXT.
592
593          *LOWER BOUNDARIES ARE OPTIONAL. *IF OMITTED
594          THEY ARE ASSUMED TO BE 1.
595          *IF A LOWER BOUNDARY IS OMITTED, THE CORRESPONDING
```

```
596           COLON SHOULD BE OMITTED TOO.

                                                                                                                    CD
597   UNIT    CD
598   BACK    CC
599   WRITE   *D*C*L ANAME (LBOUND1*:UBOUND1,LBOUND2*:UBOUND2,...) ATTR;
600   AT      314
601   WRITE   *D*C*L *B*O*A*R*D(1*:8,1*:8);

------------------------------------------------------------------------------------------------------------------- MUL-DRILL1
603           *D*C*L *B*O*A*R*D(1*:8,8);
604           *D*C*L *B*O*A*R*D(8,1*:8);
605           *D*C*L *B*O*A*R*D(8,8);
606   AT      809
607   WRITE   *THE FOUR *D*E*C*L*A*R*E STATEMENTS ABOVE DEFINE THE
608           SAME ARRAY, *B*O*A*R*D, AS A TWO DIMENSIONAL ARRAY
609           WITH 64 ELEMENTS IN 8 ROWS AND 8 COLUMNS. *THE
610           LOWER BOUNDARY FOR THE ROWS (COLUMNS, TOO) IS 1,
611           AND THE UPPER BOUNDARY (ROWS AND COLUMNS) IS 8.
612                    *B*O*A*R*D
613   CALC    HEREY#=20*HEREY*+72+ROWS#COLS#4+SISX#2+SISY#2
614   JOIN    DRAW

                                                                                                                    CE
615   UNIT    CE
616   BACK    CD
617   WRITE   *D*C*L ANAME (LBOUND1*:UBOUND1,LBOUND2*:UBOUND2,...) ATTR;
618   AT      710
619   WRITE   *IF WE WANT TO STORE THE HOURS THAT A PILOT FLEW
620           EVERY WEEK OF EVERY MONTH FOR THE YEARS 1971 TO
621           1973, WE WOULD NEED A THREE-DIMENSIONAL ARRAY
622           WITH 3 ROWS (3 YEARS), 12 COLUMNS (12 MONTHS), AND
623           5 PLANES (5 WEEKS PER MONTH). *THE STATEMENT*;
624
625               *D*C*L *H*O*U*R*S(3,12,5);
626
627           WOULD CREATE SUCH AN ARRAY. *BUT IF WE DECLARE IT AS*;
628
629               *D*C*L *H*O*U*R*S(1971*:1973,12,5);
630
631           WE CAN REFER TO AN ELEMENT AS *H*O*U*R*S(1972,7,1) AS
632           THE HOURS HE SPENT FLYING THE FIRST WEEK OF *JULY
633           IN 1972.
634   NEXT    ENDTOPIC

                                                                                                                    DRILL1
635   UNIT    DRILL1
636   AT      215
637   WRITE   *A*R*R*A*Y *D*E*C*L*A*R*A*T*I*O*N *D*R*I*L*L
638   FIGURE  214,114,138,238,214
639   AT      405
640   WRITE   *HERE YOU CAN PRACTICE DECLARATIONS OF THE FORMS*;
641           *D*C*L ANAME$$$$$*6*6*6*6*6 ( BOUND1$$$$$$*6*6*6*6*6*6,BOUND2$$$$$$*6*6*6*6*6*6,
642   WRITE   ...);   OR
643           *D*C*L ANAME$$$$$*6*6*6*6*6 ( LBOUND1$$$$$$$*6*6*6*6*6*6*6*:UBOUND$$$$$*6*6*6*6*6*6,
```

```
        LESSON PL1ARRAY      AT     17.21.26.    ON     06/13/73

644   WRITE   LBOUND2≤≤≤≤≤≤≤≤+6+6+6+6+6+6+6+6+UBOUND2≤≤≤≤≤≤≤≤+6+6+6+6+6+6+6,...)!
645
646           +USEFUL KEYS+!
647           +B+A+C+K  TO GO TO THE CONTENTS PAGE.
648           +N+E+X+T  AFTER ENTERING THE NAME OF THE ARRAY,
649                 AFTER ENTERING THE DECLARATION, AND
650                 AFTER A SUCCESSFUL DECLARATION TO
651                 DECLARE AGAIN.
652           +L+A+B   TO CHANGE THE NAME OF THE ARRAY.
653           +E+R+A+S+E TO MAKE CORRECTIONS.
654           +E+D+I+T  (ABOVE +L+A+B) PUSH IT ONCE TO SAVE YOUR ANSWER!
655                 EVERY PUSH WILL COPY A PART OF THE SAVED ANSWER
656                 SO YOU CAN CHANGE IT. +S+H+I+F+T-+E+D+I+T COPIES IT ALL.
657   FIGURE  3004,2704,2760,3060,3004
658   AT      2305
659   WRITE   +NAME OF THE ARRAY+!
                                                                              CHGE
660   ENTRY   CHGE
661   BACK    DECLARAT
662   AT      2324
663   ERASE   150
664   ARROW   2324
665   LONG    30
666   STORFA  NAME,LNAME+JCOUNT
667   OK
668   JUDGE   JCOUNT=0,IGNORE,OK
669   WRITE   +TYPE YOUR DECLARATION.
670   INHIBIT ERASE
671   JUMP    DCL

                                                                              DCL
672   UNIT    DCL
673   NEXT    DCL
674   BACK    DECLARAT
675   LAB     CHGE
676   BASE
677   INHIBIT ERASE
678   AT      2905
679   ERASE   55
680   AT      3205
681   ERASE   60
682   ARROW   2905
683   CALC    ELEM+1+DIM+AOUT+0
684   LONG    50
685   PUTD    ,/,=,
686   PUTD    ,+1,/,
687   MATCH   N98,DCL,DECLARE,+D+C+L,+D+E+C+L+A+R+E

 ------------------------------------------------------------------------------------------------------------------------ DR-INITIAL
689   JUDGE   N98,NO,CONTINUE
690   WRITE   +MISSING WORD +D+E+C+L+A+R+E.
691   STORFA  N91,N90+JCOUNT
692   OPEN    N100
693   OK
694   CALC    N99+ICOUNT
```

```
695  FIND    LP,N100,N99,PTR
696  WRITEC  PTR,*MISSING LEFT PARENTHESIS...
697  JOIN    PTR,NOT,X
698  SEARCH  NAME,LNAME,N91,N90,N98
699  WRITEC  N98,*MISSING ARRAY NAME...
700  JOIN    N98,NOT,NONE

                                                                        NOT
701  UNIT    NOT
702  JUDGE   NO
703  NO

                                                                        NONE
704  UNIT    NONE
705  JUDGE   CONTINUE
706  CLOSE   N100,PTR
707  BUMP
708  ANSVA   NAME,LNAME
709  CALC    AQUI*100+PTR*ELEM*1*DIM*0
                                                                BOUND
710  ENTRY   BOUND
711  FIND    CM,N(AQUI),100*N99-AQUI,PTR
712  CALCS   PTR,FLAG*-1,0
713  GOTO    PTR,X,BOUND1
714  FIND    RP,N(AQUI),100*N99-AQUI,PTR
715  WRITEC  PTR,*MISSING RIGHT PARENTHESIS...
716  GOTO    PTR,NOT,BOUND1

                                                                        BOUND1
717  UNIT    BOUND1
718  FIND    CL,N(AQUI),PTR,FLAG1
719  GOTO    FLAG1,BOUND2,X
720  JUDGE   RE.JUDGE
721  CLOSE   N(AQUI),FLAG1
722  STORE   B1
723  WRITE   *MISSING A NUMBER.
724  CLOSE   N(AQUI*FLAG1*1),PTR-FLAG1-1
725  STORE   B2
726  WRITE   *MISSING A NUMBER.
727  GOTO    NOT
728  OK
729  WRITEC  B1>B2,*LOWER BOUNDARY LARGER THAN UPPER...
730  GOTO    B1>B2,NOT,X
                                                                MAQUI
731  ENTRY   MAQUI
732  BREAK
733  JUDGE   RE.JUDGE
734  OK
735  CALC    AQUI*AQUI*PTR*1
736  CALCS   FLAG1,ELEM*ELEM*B1,ELEM*(B2*1-B1)
737  ADD1    DIM
738  GOTO    FLAG,BOUND3,BOUND

                                                                        BOUND2
739  UNIT    BOUND2
```

```
740  JUDGE   REJUDGE
741  CLOSE   N(AQUI),PTR
742  STORE   B1
743  WRITE   *MISSING A NUMBER.
744  GOTO    NOT
745  OK
746  WRITEC  B1>0,**BOUNDARY SMALLER THAN 0.
747  GOTO    B1>0,MAQUI,NOT

                                                                        ROUND3
748  UNIT    ROUND3
749  JUDGE   RE JUDGE
750  CLOSE   N(AQUI),100*N99-AQUI
751  PUTO    *),/,
752  PUTO    ,*,SC,
753  ANSWER  SC
754  WRITE   *P*E*R*F*E*C*T*.  *YOUR ARRAY HAS
755  CALC    N30=LOG(ELEM)*.50001
756  SHOW    ELEM,N30
757  WRITEC  ELEM=1, ELEMENT IN, ELEMENTS IN
758  SHOW    DIM,2
759  WRITEC  DIM=1, DIMENSION,, DIMENSIONS.
760  NO
761  WRITE   *MISSING SEMICOLON.

                                                                        INITIAL
762  UNIT    INITIAL
763  BACK    DECLARAT
764  WRITE   *I*N*I*T*I*A*L *A*T*T*R*I*B*U*T*E
765  AT      1010
766  WRITE   *THE *I*N*I*T*I*A*L ATTRIBUTE IS A *P*L/1 ATTRIBUTE
767          THAT ALLOWS US TO INITIALIZE THE ELEMENTS
768          OF AN ARRAY, WITHOUT USING AN ASSIGNMENT
769          STATEMENT.
770          *THE FORM OF THE *I*N*I*T*I*A*L ATTRIBUTE IS*!
771  AT      1715
772  WRITE   *I*N*I*T*I*A*L ( ITEM1<<<<<<*6*6*6*6*6*,ITEM2<<<<<<*6*6*6*6*6*,... )
773                      OR
774             *I*N*I*T ( ITEM1<<<<<<*6*6*6*6*6,ITEM2<<<<<<*6*6*6*6*6,... )
775  AT      2210
776  WRITE   ITEM1<<<<<<*6*6*6*6*6,ITEM2<<<<<<*6*6*6*6*6,... ARE INITIALIZATION
777          ITEMS AND MUST BE SEPARATED BY COMMAS.

                                                                        DA
778  UNIT    DA
779  BACK    INITIAL
780  WRITE   *I*N*I*T(ITEM1,ITEM2,...)
781  AT      510
782  WRITE   *THE INITIALIZATION ITEMS CAN BE OF THREE TYPES*!
783
784          A) *A *P*L/1 CONSTANT (4, 9.75, 2*E+03, *7STRING*7).
785             *THE ELEMENT IS SET TO THE CONSTANT.
786          B) *AN ASTERISK (*).
787             *THE ELEMENT IS NOT INITIALIZED.
788          C) *AN ITERATION FACTOR*!
```

```
789             *A POSITIVE DECIMAL INTEGER (FACTOR) IN

 ------------------------------------------------------------------------------------------------------------------------ INITIAL
791             PARENTHESES FOLLOWED BY A *P*L/1 CONSTANT
792             OR AN ASTERISK OR ANOTHER LIST OF
793             INITIALIZATION ITEMS*;
794                         (FACTOR) CONSTANT
795                         (FACTOR) *
796                         (FACTOR) (ITEM1,ITEM2,...)
797
798
799           *THE LAST FORM ALLOWS NESTING OF ITERATION FACTORS.
800
801
802           *IF THERE ARE MORE INITIALIZATION ITEMS THAN ARRAY
803           ELEMENTS, THE REMAINING ITEMS WILL BE IGNORED.
804           *IF THERE ARE MORE ELEMENTS THAN ITEMS, THE REMAINING
805           ELEMENTS WILL NOT BE INITIALIZED.

                                                                                                                        DB
806   UNIT    DB
807   BACK    DA
808   ERASE
809   AT      1020
810   WRITE   *IN THE FOLLOWING EXAMPLES
811           ////// STANDS FOR GARBAGE,
812           I.E. THE ELEMENT IS NOT
813           INITIALIZED.
814
815           *PAY SPECIAL ATTENTION TO THE
816           ORDER OF THE INITIALIZATION.

                                                                                                                        DC
817   UNIT    DC
818   BACK    DB
819   ERASE
820   AT      205
821   WRITE   *D*C*L *A(5) *I*N*I*T(1,2,3,4,5)*;
822                  *A
823   AT      415
824   WRITE   1*A(1)
825           2*A(2)
826           3*A(3)
827           4*A(4)
828           5*A(5)
829   CALC    HEREY#376+HEREY#66+SISX#1+ROWS#4+SISY#COLS#1
830   JOIN    DRAW
831   INHERIT ERASE

                                                                                                                        DCA
832   UNIT    DCA
833   BACK    DB
834   AT      232
835   WRITE   *D*C*L *B(5) *I*N*I*T(1,2,3)*;
836                    *B
```

```
A37  AT       436
A38  WRITE    ↑B(1)      1
A39           ↑B(2)      2
A40           ↑B(3)      3
A41           ↑B(4)//////
A42           ↑B(5)//////
A43  CALC     HEREX=312
A44  JOIN     DRAW
A45  INHIBIT ERASE

                                                                    DCB
A46  UNIT     DCB
A47  BACK     DB
A48  AT       1005
A49  WRITE    ↑D↑C↑L ↑C(5) ↑I↑N↑I↑T(1,2,3,4,5,6,7);
A50                  ↑C
A51  AT       1215
A52  WRITE    1↑C(1)
A53           2↑C(2)
A54           3↑C(3)
A55           4↑C(4)
A56           5↑C(5)
A57  CALC     HEREY=248↑HEREX=72
A58  JOIN     DRAW
A59  INHIBIT ERASE

                                                                    DCC
A60  UNIT     DCC
A61  BACK     DB
A62  AT       1036
A63  WRITE    ↑D↑C↑L ↑D(5);
A64                 ↑D
A65           ↑D(1)//////
A66           ↑D(2)//////
A67           ↑D(3)//////
A68           ↑D(4)//////
A69           ↑D(5)//////
A70  CALC     HEREX=312
A71  JOIN     DRAW
A72  INHIBIT ERASE

                                                                    DCD
A73  UNIT     DCD
A74  BACK     DB
A75  AT       1805
A76  WRITE    ↑D↑C↑L ↑E(5) ↑I↑N↑I↑T((5)1);
A77                  ↑E
A78  AT       2015
A79  WRITE    1↑E(1)
A80           1↑E(2)
A81           1↑E(3)
A82           1↑E(4)
A83           1↑E(5)
A84  CALC     HEREY=120↑HEREX=72
A85  JOIN     DRAW
```

```
886   INHIBIT ERASE

                                                                                                                    DCE
887   UNIT    DCE
888   BACK    DB
889   AT      1830
890   WRITE   *D*C*L *F(5) *I*N*I*T(1,2,*,4,5);
891                       *F
892   AT      2036
893   WRITE   *F(1)      1
894           *F(2)      2
895           *F(3)//////
896           *F(4)      4
897           *F(5)      5
898   CALC    HEREX+312
899   JOIN    DRAW
900   INHIBIT ERASE

                                                                                                                    DCF
901   UNIT    DCF
902   BACK    DH
903   AT      2605
904   WRITE   *D*C*L *G(5) *I*N*I*T((2)((2)1,0));          *G
905           SAME AS*!
906           *D*C*L *G(5) *I*N*I*T(1,1,0,1,1,0);
907   AT      2736

------------------------------------------------------------------------------------------------------------------- INITIAL
909   WRITE   *G(1)      1
910           *G(2)      1
911           *G(3)      0
912           *G(4)      1
913           *G(5)      1
914   CALC    HERE+*90
915   JOIN    DRAW

                                                                                                                    DD
916   UNIT    DD
917   BACK    DC
918   ERASE
919   AT      101
920   WRITE   *I*N*I*T*I*A*L *A*T*T*R*I*B*U*T*E
921   AT      1010
922   WRITE   *THE *I*N*I*T*I*A*L ATTRIBUTE IS USED TO GIVE INITIAL
923           VALUES TO THE ELEMENTS OF AN ARRAY, AND CAN BE
924           USED ONLY ONCE*IN THE *D*E*C*L*A*R*E STATEMENT.
925
926           *THE *G*E*T AND THE ASSIGNMENT STATEMENTS ARE USED
927           TO GIVE INITIAL VALUES TO THE ELEMENTS OF AN
928           ARRAY OR TO CHANGE THESE VALUES, AND MAY BE
929           USED SEVERAL TIMES IN A PROGRAM.
930
931           *WE SHOULD NEVER ASSUME THAT AN ARRAY ELEMENT
932           HAS A PARTICULAR VALUE, UNLESS EXPLICITLY
933           SPECIFIED BY THE *I*N*I*T*I*A*L ATTRIBUTE OR A *G*E*T
```

```
934           OR ASSIGNMENT STATEMENT.

                                                                        DE
935   UNIT    DE
936   BACK    DD
937   ERASE
938   AT      101
939   WRITE   ↑I↑N↑I↑T↑I↑A↑L ↑A↑T↑T↑R↑I↑B↑U↑T↑E
940   AT      1015
941   WRITE   ↑THE INITIALIZATION OF MULTIDIMENSIONAL
942           ARRAYS IS PERFORMED IN ↑7ROW MAJOR ORDER.↑7
943           WHICH MEANS THAT THE RIGHTMOST SUBSCRIPT
944           IS VARIED MOST RAPIDLY AND THE LEFTMOST
945           SUBSCRIPT IS VARIED LEAST RAPIDLY.
946
947           ↑THE VALUES FOR THE INITIALIZATION ARE
948           TAKEN ONE BY ONE FROM LEFT TO RIGHT FROM
949           THE LIST OF INITIALIZATION ITEMS FOUND
950           IN THE ↑I↑N↑I↑T↑I↑A↑L ATTRIBUTE.

                                                                        DF
951   UNIT    DF
952   BACK    DD
953   ERASE
954   AT      312
955   WRITE   ↑D↑C↑L ↑A↑A(2,5) ↑I↑N↑I↑T(1,2,3,4,5,6,7,8,9,10);
956                         ↑A↑A
957   AT      510
958   WRITE   ↑A↑A(1,1)     1     2     3     4     5↑A↑A(1,5)
959           ↑A↑A(2,1)     6     7     8     9    10↑A↑A(2,5)
960   CALC    HEREY,304↑HEREX,123↑COLS,3↑ROWS,2
961   JOIN    DRAW
962   INHIBIT ERASE

                                                                        DFA
963   UNIT    DFA
964   BACK    DD
965   AT      912
966   WRITE   ↑D↑C↑L ↑B↑B(5,2) ↑I↑N↑I↑T↑I↑A↑L(1,2,3,4,5,6,7,8,9,10);
967                         ↑B↑B
968   AT      1119
969   WRITE   ↑B↑B(1,1)     1     2↑B↑B(1,2)
970           ↑B↑B(2,1)     3     4↑B↑B(2,2)
971           ↑B↑B(3,1)     5     6↑B↑B(3,2)
972           ↑B↑B(4,1)     7     8↑B↑B(4,2)
973           ↑B↑B(5,1)     9    10↑B↑B(5,2)
974   CALC    HEREY,136↑HEREX,195↑ROWS,3↑COLS,2
975   JOIN    DRAW
976   INHIBIT ERASE

                                                                        DFB
977   UNIT    DFB
978   BACK    DD
979   AT      1812
980   WRITE   ↑D↑C↑L ↑C↑C(3,5) ↑I↑N↑I↑T(1,2,(5)3.8,(6)*,15);
```

```
                        ↑C↑C
AT      2010
WRITE   ↑C↑C(1,1)      1     2     3     3     3↑C↑C(1,5)
        ↑C↑C(2,1)      3     3     8/////////////↑C↑C(2,5)
        ↑C↑C(3,1)//////////////////////////    15↑C↑C(3,5)
CALC    HEREY=64↑HEREX=123↑COLS=2↑ROWS=3
JOIN    DRAW
INHIBIT ERASE

                                                                                                    DFC
UNIT    DFC
BACK    DD
AT      2512
WRITE   ↑D↑C↑L ↑D↑D(3,5) ↑I↑N↑I↑T((3)(1,2,3,*,5));
                        ↑D↑D
AT      2710
WRITE   ↑D↑D(1,1)      1     2     3//////      5↑D↑D(1,5);
        ↑D↑D(2,1)      1     2     3//////      5↑D↑D(2,5);

 ------------------------------------------------------------------------------------------------------ INITIAL-DR
        ↑D↑D(3,1)      1     2     3//////      5↑D↑D(3,5);
CALC    HERE=80
JOIN    DRAW

                                                                                                    DG
UNIT    DG
BACK    DF
ERASE
AT      608
WRITE   ↑D↑C↑L ↑C↑A↑R↑S(2,3) ↑C↑H↑A↑R(11) ↑I↑N↑I↑T(↑7↑F↑O↑R↑D↑7,↑7↑P↑I↑N↑T↑O↑7,↑7↑C↑A↑D↑I↑L↑L↑A↑C↑7
        ,↑7↑F↑I↑A↑T↑7,↑7↑T↑O↑Y↑O↑T↑A↑7,↑7↑R↑O↑L↑L↑S ↑R↑O↑Y↑C↑E↑7);
                        ↑C↑A↑R↑S
AT      904
WRITE   ↑C↑A↑R↑S(1,1)↑F↑O↑R↑D       ↑P↑I↑N↑T↑O      ↑C↑A↑D↑I↑L↑L↑A↑C   ↑C↑A↑R↑S(1,3)
        ↑C↑A↑R↑S(2,1)↑F↑I↑A↑T       ↑T↑O↑Y↑O↑T↑A     ↑R↑O↑L↑L↑S ↑R↑O↑Y↑C↑E↑C↑A↑R↑S(2,3)
CALC    HEREY=256↑HEREX=101↑SISX=8↑COLS=1↑ROWS=2
JOIN    DRAW
INHIBIT ERASE

                                                                                                    DGA
UNIT    DGA
BACK    DF
AT      1508
WRITE   ↑D↑C↑L ↑C↑A↑R$(2,3) ↑C↑H↑A↑R(11) ↑I↑N↑I↑T((3)(↑7↑C↑A↑D↑I↑L↑L↑A↑C↑7),
            (3)(↑7↑R↑O↑L↑L↑S ↑R↑O↑Y↑C↑E↑7));
                        ↑C↑A↑R$
AT      1806
WRITE   ↑C↑A↑R$(1,1)↑C↑A↑D↑I↑L↑L↑A↑C   ↑C↑A↑D↑I↑L↑L↑A↑C   ↑C↑A↑D↑I↑L↑L↑A↑C   ↑C↑A↑R$(1,3)
        ↑C↑A↑R$(2,1)↑R↑O↑L↑L↑S ↑R↑O↑Y↑C↑E↑R↑O↑L↑L↑S ↑R↑O↑Y↑C↑E↑R↑O↑L↑L↑S ↑R↑O↑Y↑C↑E↑C↑A↑R$(2,3)
CALC    HEREY=224
JOIN    DRAW
INHIBIT ERASE

                                                                                                    DGB
UNIT    DGB
```

```
        LESSON PL¡ARRAY      AT     17.21.26.    ON     06/13/73

1027  BACK    DF
1028  AT      2408
1029  WRITE   ↑D↑C↑L ↑C↑A↑R(2,3) ↑I↑N↑I↑T((5)*,↑7↑V↑O↑L↑K↑S↑W↑A↑G↑E↑N↑7) ↑C↑H↑A↑R(11)!
1030                              ↑C↑A↑R
1031  AT      2607
1032  WRITE   ↑C↑A↑R(1,1)//////////////////////////////////↑C↑A↑R(1,3)
1033          ↑C↑A↑R(2,1)/////////////////////////↑V↑O↑L↑K↑S↑W↑A↑G↑E↑N ↑C↑A↑R(2,3)
1034  CALC    HEREY←96
1035  JOIN    DRAW

                                                                                                    DH
1036  UNIT    DH
1037  BACK    DG
1038  AT      505
1039  WRITE   ↑D↑C↑L ↑S↑T↑O↑C↑K(2,2,3) ↑C↑H↑A↑R(10) ↑I↑N↑I↑T(↑7↑D↑I↑O↑D↑E↑7,↑71&U↑7,*,
1040             (2)(↑7↑T↑R↑A↑N↑S↑I↑S↑T↑O↑R↑7),(2)(↑7100↑7,*),↑7↑A ↑M↑O↑U↑S↑E↑7,
1041  WRITE   ↑7↑I↑I↑I↑I↑I↑I↑.↑.↑.↑7)!
1042  AT      1515
1043  WRITE   ↑S↑T↑O↑C↑K(1,1,3) //////////100          ↑S↑T↑O↑C↑K(1,2,3)
1044          ↑S↑T↑O↑C↑K(2,1,3) ////////////////////  ↑S↑T↑O↑C↑K(2,2,3)
1045  AT      1809
1046  WRITE   ↑S↑T↑O↑C↑K(1,1,2) 140        ↑T↑R↑A↑N↑S↑I↑S↑T↑O↑R ↑S↑T↑O↑C↑K(1,2,2)
1047          ↑S↑T↑O↑C↑K(2,1,2) 100        ↑I↑I↑I↑I↑I↑I↑.↑.↑.  ↑S↑T↑O↑C↑K(2,2,2)
1048  AT      2103
1049  WRITE   ↑S↑T↑O↑C↑K(1,1,1) ↑D↑I↑O↑D↑E      ↑T↑R↑A↑N↑S↑I↑S↑T↑O↑R ↑S↑T↑O↑C↑K(1,2,1)
1050          ↑S↑T↑O↑C↑K(2,1,1) ////////////↑A ↑M↑O↑U↑S↑E     ↑S↑T↑O↑C↑K(2,2,1)
1051  CALC    HEREY←56←HEREX←206←SISX←8←COLS←2
1052  JOIN    DRAW
1053  CALC    HEREY←56←HEREX←168
1054  JOIN    DRAW
1055  CALC    HEREY←56←HEREX←120
1056  JOIN    DRAW
1057  LINE    119,192,215,288
1058  LINE    279,160,375,256
1059  NEXT    ENDTOPIC

                                                                                                    DRILL2
1060  UNIT    DRILL2
1061  ERASE
1062  AT      215
1063  WRITE   ↑I↑N↑I↑T↑I↑A↑L ↑A↑T↑T↑R↑I↑B↑U↑T↑E ↑D↑R↑I↑L↑L
1064  FIGURE  214,114,138,238,214
1065  SETPERM 12
1066  CALC    HEREY←222←HEREX←2←N48←52←N49←41←SISX←6←ROWS←1←COLS←1←SISY←2
1067          N47←1
1068  AT      605
1069  WRITE   ↑FILL IN THE ARRAY WITH THE CORRESPONDING VALUES
1070          ACCORDING TO THE FOLLOWING STATEMENT↑!
1071
1072          ↑D↑C↑L ↑A(4,3) ↑I↑N↑I↑T(  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  )!
1073  AT      920
1074  RANDP   N50

                                                                                                    LOOP
1075  ENTRY   LOOP
1076  CALC    N(N49)←N50←10
```

```
1077  SHOW    N(N49),3

-------------------------------------------------------------------------------- #RILL 2-ENT
1079  ADD1    N49
1080  RAND#   N50
1081  GOTO    N50=0,X.LOOP
1082  JOIN    DRAW
1083  AT      2019
1084  WRITE   *NUMBER OF GUESSES*!
1085  MOD#ERM
1086  AT      2310
1087  WRITE   *PRESS *D*A*T*A TO GUESS A DIFFERENT ELEMENT.
1088          *PRESS *L*A*B FOR A DIFFERENT *D*E*C*L*A*R*E STATEMENT.
1089          *PRESS *B*A*C*K TO GO TO THE CONTENTS PAGE.
1090  INHIBIT ERASE
1091  JUM#    GUESS

                                                                                GUESS
1092  UNIT    GUESS
1093  DATA    GUESS
1094  LAB     DRILL2
1095  BACK    DECLARAT
1096  NEXT    DRILL2
1097  INHIBIT ERASE
1098  MODE    REWRITE
1099  MOD#ERM
1100  AT      N48
1101  WRITE
1102  RAND#   N50
1103  GOTO    N50=0,DONE,X
1104  CALCS   N50,N48*0+0,1215,1225,1235,1415,1425,1435,1615,1625
1105          1635,1815,1825,1835
1106  ARROW   N48
1107  LONG    2
1108  FORCE   LONG
1109  STORE   N49
1110  ADD1    N47
1111  OK
1112  AT      2037
1113  SHOW    N47,2
1114  ADD1    N47
1115  JUDGE   N(50,N50)=N49,OK,IGNORE
1116  REMOVE  N50
1117  JUM#    GUESS

                                                                                DONE
1118  UNIT    DONE
1119  AT      3030
1120  WRITE   *D*O*N*E*.*.*.

                                                                                ENTIRE
1121  UNIT    ENTIRE
1122  BACK    CONTENTS
1123  ERASE
1124  AT      101
```

```
1125  WRITE   *E*N*T*I*R*E *A*R*R*A*Y *O*P*E*R*A*T*I*O*N*S
1126  AT      510
1127  WRITE   *WHEN WE USE THE NAME OF AN ARRAY
1128          WITHOUT SUBSCRIPTS IN A PROGRAM,
1129          WE ARE REFERENCING THE ENTIRE ARRAY.
1130
1131          *WE CAN USE ENTIRE ARRAYS*!
1132  AT      1210
1133  WRITE   1. *IN ASSIGNMENT STATEMENTS OF THE FORM*!
1134             ANAME<<<<<*6*6*6*6*6=<CONSTANT,VARIABLE,OR ARRAY EXPRESSION>!
1135
1136          2. *WITH BUILT-IN ARRAY FUNCTIONS*!
1137             FCNAME<<<<<*6*6*6*6*6*6(ANAME<<<<*6*6*6*6*6).
1138
1139          3. *FOR *INPUT AND *OUTPUT OF ARRAYS.
1140             (SEE *INPUT/*OUTPUT OF ARRAYS)
1141  AT      2210
1142  WRITE   *ALL ARRAYS IN AN ASSIGNMENT STATEMENT MUST
1143          HAVE THE SAME NUMBER OF DIMENSIONS AND THE
1144          SAME BOUNDARY LIMITS.
                                                                    EA
1145  UNIT    EA
1146  BACK    ENTIRE
1147  ERASE
1148  AT      101
1149  WRITE   *A*R*R*A*Y *A*S*S*I*G*N*M*E*N*T *S*T*A*T*E*M*E*N*T
1150  AT      505
1151  WRITE   *D*C*L (*A(4) *I*N*I*T(1,2,3,4),
1152             *B(4),*C(4))!
1153  AT      533
1154  WRITE   *CREATES ARRAYS *A, *B, AND *C,
1155          AND INITIALIZES ARRAY *A.
1156  AT      2412
1157  WRITE   *A              *B              *C
1158             1
1159             2
1160             3
1161             4
1162  CALC    HEREY*40*HEREX*66*SISX*2*ROWS*3*COLS*SISY*1
1163  PACK    STRLEN,STRING,//////
1164  JOIN    DRAW
1165  CALC    HEREX*192
1166  JOIN    DRAW
1167  JOIN    FILL
1168  CALC    HEREX*312
1169  JOIN    DRAW
1170  JOIN    FILL
1171  INMIDIT ERASE
                                                                    EAA
1172  UNIT    EAA
1173  BACK    ENTIRE
1174  AT      805
1175  WRITE   *C = 1!
```

```
1176  AT      833
1177  WRITE   *SETS ALL ELEMENTS OF
1178          ARRAY *C TO 1.
1179  MODE    ERASE
1180  JOIN    FILL
1181  MODE    WRITE
1182  PACK    STRING.     1
1183  JOIN    FILL
1184  INHIBIT ERASE

                                                                                EAB
1185  UNIT    EAB
1186  BACK    ENTIRE
1187  AT      1105
1188  WRITE   *B = *A + *C!

  ------------------------------------------------------------------------------------------ ENTIRE
1190  AT      1133
1191  WRITE   *STORES IN EVERY ELEMENT OF
1192          *B, THE SUM OF THE CORRESPONDING
1193          ELEMENTS OF ARRAYS *A AND *C.
1194  PACK    STRING.//////
1195  CALC    HERE*192
1196  MODE    ERASE
1197  JOIN    FILL
1198  MODE    WRITE
1199  AT      2530
1200  WRITE   2
1201          3
1202          4
1203          5
1204  INHIBIT ERASE

                                                                                EAC
1205  UNIT    EAC
1206  BACK    ENTIRE
1207  AT      1505
1208  WRITE   *A=2**B!
1209  AT      1533
1210  WRITE   *STORES IN EVERY ELEMENT OF *A,
1211          TWO TIMES THE CONTENTS OF THE
1212          CORRESPONDING ELEMENT OF *B.
1213  AT      2515
1214  MODE    ERASE
1215  WRITE   1
1216          2
1217          3
1218          4
1219  MODE    WRITE
1220  AT      2514
1221  WRITE    4
1222           6
1223           8
1224          10
1225  INHIBIT ERASE
```

EAD

```
1226  UNIT    EAD
1227  BACK    ENTIRE
1228  AT      1905
1229  WRITE   ↑C = ↑A**↑B!
1230  AT      1933
1231  WRITE   ↑STORES IN EVERY ELEMENT
1232          OF ↑C, THE PRODUCT OF THE
1233          CORRESPONDING ELEMENTS
1234          OF ↑A AND ↑B.
1235  CALC    HERE X↑312
1236  PACK    STRING,      1
1237  MODE    ERASE
1238  JOIN    FILL
1239  MODE    WRITE
1240  AT      2544
1241  WRITE    8
1242          18
1243          32
1244          50
```

FB

```
1245  UNIT    EB
1246  BACK    EA
1247  ERASE
1248  AT      101
1249  WRITE   ↑B↑U↑I↑L↑T-↑I↑N ↑A↑R↑R↑A↑Y ↑F↑U↑N↑C↑T↑I↑O↑N↑S
1250  AT      812
1251  WRITE   ↑THERE ARE SEVEN BUILT-IN FUNCTIONS FOR
1252          ARRAYS AND ARRAY EXPRESSIONS IN ↑P↑L/1↑!
1253
1254
1255          ↑H↑B↑O↑U↑N↑D, ↑L↑B↑O↑U↑N↑D, AND ↑D↑I↑M GIVE VALUES
1256          WHICH DEPEND ON THE BOUNDARIES OF THE
1257          SUBSCRIPTS OF THE ARRAY.
1258
1259          ↑S↑U↑M, AND ↑P↑R↑O↑D OPERATE ON ALL THE ELEMENTS
1260          OF THE ARRAY AS NUMBERS.
1261
1262          ↑A↑L↑L, AND ↑A↑N↑Y OPERATE ON ALL THE ELEMENTS
1263          OF THE ARRAY AS BIT-STRINGS.
```

EC

```
1264  UNIT    EC
1265  BACK    EB
1266  ERASE
1267  AT      101
1268  WRITE   ↑B↑U↑I↑L↑T-↑I↑N ↑A↑R↑R↑A↑Y ↑F↑U↑N↑C↑T↑I↑O↑N↑S
1269  AT      310
1270  WRITE   ↑D↑C↑L (↑A(-1↑!1,7),↑B(5,4,10↑!20))!
1271  AT      608
1272  WRITE   ↑H↑B↑O↑U↑N↑D(AEXP↑5↑5↑5↑6↑6↑6↑6,N<↑6)
1273           ↑H↑B↑O↑U↑N↑D RETURNS THE UPPER BOUNDARY OF THE
1274           N<↑6TH DIMENSION OF AEXP↑5↑5↑5↑6↑6↑6↑6.
```

```
1275            ↑I=↑H↑R↑O↑U↑N↑D(↑A,1)!   ↑I CONTAINS 1.
1276            ↑J=↑H↑R↑O↑U↑N↑D(↑A,2)!   ↑J CONTAINS 4.
1277            ↑K=↑H↑R↑O↑U↑N↑D(↑A,3)!   ↑K CONTAINS 20.
1278    INHIBIT ERASE

                                                                                ECA
1279    UNIT    ECA
1280    BACK    EB
1281    AT      140A
1282    WRITE   ↑L↑R↑O↑U↑N↑D(AEXP≤≤≤≤↑6↑6↑6↑6,N<↑6)
1283            ↑L↑R↑O↑U↑N↑D RETURNS THE LOWER BOUNDARY OF THE
1284            N<↑6TH DIMENSION OF AEXP≤≤≤≤↑6↑6↑6↑6.
1285            ↑L=↑L↑R↑O↑U↑N↑D(↑A,1)!   ↑L CONTAINS -1.
1286            ↑M=↑L↑R↑O↑U↑N↑D(↑A,2)!   ↑M CONTAINS 1.
1287            ↑N=↑L↑R↑O↑U↑N↑D(↑A,3)!   ↑N CONTAINS 10.
1288    INHIBIT ERASE

                                                                                ECB
1289    UNIT    ECA
1290    BACK    EB
1291    AT      220A
1292    WRITE   ↑D↑I↑M(AEXP≤≤≤≤↑6↑6↑6↑6,N≤↑6)
1293            ↑D↑I↑M RETURNS THE EXTENT OF THE N≤↑6TH DIMENSION
1294            OF AEXP≤≤≤≤↑6↑6↑6↑6, I.E. THE NUMBER OF DIFFERENT VALUES

  ------------------------------------------------------------------------------------------------------------------------------------ ENTIRE
1296            THAT THE N≤↑6TH SUBSCRIPT CAN TAKE.
1297            ↑D↑I↑M(AEXP≤≤≤≤↑6↑6↑6↑6,N≤↑6) = 1 + ↑H↑R↑O↑U↑N↑D(AEXP≤≤≤≤↑6↑6↑6↑6,N≤↑6) - ↑L↑R↑O↑U↑N↑D(AEXP≤≤≤≤↑6↑6↑6↑6,N<↑6)
1298            ↑P=↑D↑I↑M(↑A,1)!  ↑P CONTAINS 3
1299            ↑Q=↑D↑I↑M(↑B,2)!  ↑Q CONTAINS 4
1300            ↑R=↑D↑I↑M(↑B,3)!  ↑R CONTAINS 11

                                                                                FD
1301    UNIT    ED
1302    BACK    EC
1303    ERASE
1304    AT      101
1305    WRITE   ↑B↑U↑I↑L↑T-↑I↑N ↑A↑R↑R↑A↑Y ↑F↑U↑N↑C↑T↑I↑O↑N↑S
1306    CALC    HEREV=320↑HEREX=109↑ROWS=1↑COLS=2
1307    JOIN    DRAW
1308    CALC    HEREX=271↑COLS=1
1309    JOIN    DRAW
1310    AT      420
1311    WRITE   ↑A
1312            1     2
1313            3     4
1314            5     6
1315    AT      437
1316    WRITE   ↑B
1317               7
1318               8
1319               9
1320    AT      1010
1321    WRITE   ↑S↑U↑M(AEXP≤≤≤≤↑6↑6↑6↑6)
1322            ↑S↑U↑M RETURNS THE SUM OF ALL THE ELEMENTS OF AEXP≤≤≤≤↑6↑6↑6↑6.
```

```
1323            ↑G=↑S↑U↑M(↑A);          ↑G CONTAINS 21 (1↑2↑3↑4↑5↑6)
1324            ↑H=↑S↑U↑M(↑B);          ↑H CONTAINS 24 (7↑8↑9)
1325            ↑I=↑S↑U↑M(↑A(*,2));     ↑I CONTAINS 12 (2↑4↑6)
1326            ↑J=↑S↑U↑M(↑B-↑A(*,1));  ↑J CONTAINS 15 (6↑5↑4)
1327    INHIBIT ERASE
                                                                        EDA
1328    UNIT    EDA
1329    BACK    EC
1330    AT      1810
1331    WRITE   ↑P↑R↑O↑D(AEXP≤≤≤≤↑6↑6↑6↑6)
1332             ↑P↑R↑O↑D RETURNS THE PRODUCT OF ALL THE ELEMENTS
1333             OF AEXP≤≤≤≤↑6↑6↑6↑6.
1334             ↑K=↑P↑R↑O↑D(↑A);        ↑K CONTAINS 720 (1≠2≠3≠4≠5≠6)
1335             ↑L=↑P↑R↑O↑D(↑B);        ↑L CONTAINS 504 (7≠8≠9)
1336             ↑M=↑P↑R↑O↑D(↑A(*,2));   ↑M CONTAINS  48 (2≠4≠6)
1337             ↑N=↑P↑R↑O↑D(↑B-↑A(*,1)); ↑N CONTAINS 120 (6≠5≠4)
                                                                        EE
1338    UNIT    EE
1339    BACK    ED
1340    WRITE   ↑B↑U↑I↑L↑T-↑IN ↑A↑R↑R↑A↑Y ↑F↑U↑N↑C↑T↑I↑O↑N↑S
1341    CALC    HEREX↑10B↑ROWS↑2↑COLS↑2
1342    JOIN    DRAW
1343    CALC    HEREX↑271↑COLS↑1
1344    JOIN    DRAW
1345    AT      418
1346    WRITE     ↑A
1347            000   100
1348            001   101
1349            010   110
1350            011   111
1351    AT      438
1352    WRITE   ↑B
1353             00
1354             01
1355             10
1356             11
1357    AT      910
1358    WRITE   ↑CONTENTS OF ↑A AND ↑B ARE BIT-STRINGS.
1359
1360
1361            ↑A↑L↑L(AEXP≤≤≤≤↑6↑6↑6↑6)
1362             ↑A↑L↑L RETURNS A BIT-STRING CONTAINING A 1 IN ANY
1363             POSITION ONLY IF ALL THE ELEMENTS OF AEXP≤≤≤≤↑6↑6↑6↑6 HAVE
1364             A 1 IN THE SAME POSITION, AND A 0 OTHERWISE.
1365             ↑G=↑A↑L↑L(↑A);          ↑G CONTAINS 000↑B.
1366             ↑H=↑A↑L↑L(↑B);          ↑H CONTAINS  00↑B.
1367             ↑I=↑A↑L↑L(↑A(*,2));     ↑I CONTAINS 100↑B.
1368             ↑J=↑A↑L↑L(↑A(*,1)-=↑B); ↑J CONTAINS 000↑B
1369    INHIBIT ERASE
                                                                        EEA
1370    UNIT    EEA
1371    BACK    ED
```

```
1372  AT      211^
1373  WRITE   ^A^N^Y(AEXP<<<<<^6^6^6^6)
1374            ^A^N^Y RETURNS A BIT-STRING CONTAINING A 1 IN ANY
1375            POSITION IF ANY OF THE ELEMENTS OF AEXP<<<<^6^6^6^6 HAS A
1376            1 IN THE SAME POSITION, AND A 0 OTHERWISE.
1377            ^K=^A^N^Y(^A);          ^K CONTAINS 111^B.
1378            ^L=^A^N^Y(^B);          ^L CONTAINS  11^B.
1379            ^M=^A^N^Y(^A(*,1));     ^M CONTAINS 011^B.
1380            ^N=^A^N^Y(^A(*,2)-^^B); ^N CONTAINS 110^B.
1381  NEXT    ENDTOPIC
```

PARTS

```
1383  UNIT    PARTS                                                                    PARTS
1384  BACK    CONTENTS
1385  ERASE
1386  AT      101
1387  WRITE   ^0^P^E^R^A^T^I^0^N^S ^W^I^T^H ^P^A^R^T^S ^0^F ^A^N ^A^R^R^A^Y(A)
1388  AT      510
1389  WRITE   ^WE CAN REFERENCE PARTS OF AN ARRAY WITH A
1390          A SUBSCRIPTED NAME^!
1391
1392             ANAME<<<<<^6^6^6^6^6 (SUBS1<<<<<^6^6^6^6^6,SUBS2<<<<<^6^6^6^6^6,...)
1393
1394          ^IF ALL OF THE SUBSCRIPTS ARE INTEGERS, VARIABLES,
1395          OR ARITHMETIC EXPRESIONS, WE ARE REFERENCING A
1396          SINGLE ELEMENT OF THE ARRAY.
1397          ^EVERY SUBSCRIPT IS EVALUATED AND CONVERTED TO AN
1398          INTEGER, IF NECESARY, AND MUST LIE WITHIN THE
1399          SUBSCRIPT BOUNDARIES.
1400  INHIBIT ERASE
```

```
1401  UNIT    FA                                                                       FA
1402  BACK    CONTENTS
1403  AT      1810
1404  WRITE   ^ALL OF THE FOLLOWING REFERENCES TO AN ELEMENT
1405          OF ARRAY ^A(5,5) ARE LEGAL. (^SUPPOSE ^N=4,^M=10)
1406  AT      2215
1407  WRITE   ^A(1,1)          REFERENCES ^A(1,1)
1408          ^A(4,^N)          REFERENCES ^A(4,4)
1409          ^A((^M-1)/2,^N-2) REFERENCES ^A(4,2)
1410          ^A(^M-2*^N,7/^N)   REFERENCES ^A(2,1)
```

```
1411  UNIT    FB                                                                       FB
1412  BACK    PARTS
1413  ERASE
1414  AT      101
1415  WRITE   ^0^P^E^R^A^T^I^0^N^S ^W^I^T^H ^P^A^R^T^S ^0^F ^A^N ^A^R^R^A^Y(B)
1416  AT      513
1417  WRITE   ANAME<<<<<^6^6^6^6^6 (SUBS1<<<<<^6^6^6^6^6,SUBS2<<<<<^6^6^6^6^6,...)
1418  AT      710
1419  WRITE   ^IF ONE OR MORE OF THE SUBSCRIPTS IS AN
```

```
1420          ASTERISK (*). WE ARE REFERENCING A CROSS-
1421          SECTION OF THE ARRAY.!
1422    INHIBIT ERASE

                                                                        FBA
1423    UNIT    FBA
1424    BACK    PARTS
1425    AT      1220
1426    WRITE   *D*C*L *A*A(B,5) *C*H*A*R(6)!
1427                    *A*A
1428    CALC    HEREY=176*HEREX=104*ROWS=2*SISX=1*COLS=4*SISY=1
1429    JOIN    DRAW
1430    AT      2510
1431    WRITE   *A*A(6,*)  REFERS TO ALL THE ELEMENTS OF ROW 6
1432                    (ARRAY EXPRESSION CONTAINING 5 ELEMENTS).
1433    FIGUREF 109,206,109,226,353,226,353,206,109,206
1434    INHIBIT ERASE

                                                                        FBB
1435    UNIT    FBB
1436    BACK    PARTS
1437    MODE    ERASE
1438    FIGUREF 109,206,109,226,353,226,353,206,109,206
1439    MODE    WRITE
1440    JOIN    DRAW
1441    AT      2710
1442    WRITE   *A*A(*,2)  REFERS TO ALL THE ELEMENTS OF COLUMN 2
1443                    (ARRAY EXPRESSION CONTAINING 8 ELEMENTS).
1444    FIGUREF 157,206,209,206,209,174,157,174,157,206
1445    INHIBIT ERASE

                                                                        FBC
1446    UNIT    FBC
1447    BACK    PARTS
1448    MODE    ERASE
1449    FIGUREF 157,206,209,206,209,174,157,174,157,206
1450    MODE    WRITE
1451    JOIN    DRAW
1452    AT      2910
1453    WRITE   *A*A(*,*)  REFERS TO ALL THE ELEMENTS OF ALL THE
1454                    COLUMNS. *A*A(*,*) IS THE SAME AS *A*A
1455                    (ARRAY EXPRESSION CONTAINING 40 ELEMENTS).
1456    FIGUREF 107,172,107,308,355,308,355,172,107,172

                                                                        FC
1457    UNIT    FC
1458    BACK    FB
1459    ERASE
1460    AT      101
1461    WRITE   *OPERATIONS WITH PARTS OF AN ARRAY
1462    AT      508
1463    WRITE   *WHEN WE USE ARRAY ELEMENTS WITH CROSS-SECTIONS
1464            AND ENTIRE ARRAYS, THE COMPUTATIONS ARE
1465            PERFORMED IN ROW MAJOR ORDER.!
1466    INHIBIT ERASE
```

```
                                                                                                                    FCA
UNIT    FCA
BACK    FB
AT      90A
WRITE   *D*C*L(*A(5)**B(10,5));

        *A=*B(3,*); IS EQUIVALENT TO*;
```

PARTS-I/O

```
          *A(1)=*B(3,1);
          *A(2)=*B(3,2);
          *A(3)=*B(3,3);
INHIBIT ERASE
```

```
                                                                                                                    FCB
UNIT    FCB
BACK    FB
AT      160A
WRITE   *A(3)=*S*U*M(*B(7,*)); IS EQUIVALENT TO*;
          *A(3)=*B(7,1)+*B(7,2)+*B(7,3)+*B(7,4)+*B(7,5);
INHIBIT ERASE
```

```
                                                                                                                    FCC
UNIT    FCC
BACK    FB
AT      190A
WRITE   *B(1,*)=*B(*,1)/*A(3); IS ILLEGAL, BECAUSE *B(1,*)
        IS AN ARRAY EXPRESSION WITH 5 ELEMENTS, AND
        *B(*,1) IS AN ARRAY EXPRESSION WITH 10 ELEMENTS.
```

```
                                                                                                                    FD
UNIT    FD
BACK    FC
AT      307
WRITE   *D*C*L(*G*R*A*D*E*S(10,5)+*F*I*N*A*L(10)) *F*I*X*E*D *D*E*C(6);

                        *G*R*A*D*E*S                *F*I*N*A*L
CALC    HEREY=340+HEREX=62+ROWS=5+COLS=5
JOIN    DRAW
CALC    HEREX=327+COLS=1
JOIN    DRAW
AT      190S
WRITE   *THE FOLLOWING SECTION OF A PROGRAM CALCULATES
        THE FINAL GRADES OF THE STUDENTS IN A CLASS
        FROM THE RESULTS OF 5 EXAMS*;
AT      2415
WRITE   ....
        *D*O *I=1 *T*O 10;
          *F*I*N*A*L(*I)=*S*U*M(*G*R*A*D*E*S(*I,*))/5;
        *E*N*D;
        ....
NEXT    ENDTOPIC
```

ARRAYI/O

```
1511  UNIT    ARRAYI/O
1512  BACK    CONTENTS
1513  ERASE
1514  AT      101
1515  WRITE   *I*N*P*U*T/*O*U*T*P*U*T *O*F*F *A*R*R*A*Y*S
1516  AT      1015
1517  WRITE   *ANY OF THE THREE KINDS OF
1518          *INPUT/*OUTPUT IN *P*L/1
1519          (*L*I*S*T-DIRECTED, *D*A*T*A-DIRECTED,
1520          OR *F*O*I*T-DIRECTED) CAN BE USED
1521          TO INPUT AND OUTPUT ARRAYS
1522          OR PARTS OF AN ARRAY.
1523
1524          *INPUT/*OUTPUT OF ARRAYS IS DONE
1525          IN ROW MAJOR ORDER. I.E. THE
1526          RIGHTMOST SUBSCRIPT IS VARIED
1527          MOST RAPIDLY AND THE LEFTMOST
1528          SUBSCRIPT IS VARIED LEAST RAPIDLY.

                                                                    GA
1529  UNIT    GA
1530  BACK    ARRAYI/O
1531  ERASE
1532  AT      101
1533  WRITE   *L*I*S*T-*D*I*R*E*C*T*E*D *I*N*P*U*T
1534  AT      303
1535  WRITE   *IN *L*I*S*T-DIRECTED INPUT THE DATA ITEMS MUST BE
1536           SEPARATED BY COMMAS, OR BY ONE OR MORE BLANKS*!
1537  FIGURE  703,503,404,445
1538  FIGURE  1003*803*704*735
1539  AT      604
1540  WRITE   1,2,3 4  5   6 ,7,8 9,10,11 12,13,14 15
1541
1542
1543           16,17 18     19,20,21 22
1544  INHIBIT ERASE

                                                                    GAA
1545  UNIT    GAA
1546  BACK    ARRAYI/O
1547  AT      1205
1548  WRITE   *D*C*L(*A(6),*B(4,2),*C(5,3))!
1549  AT      1230
1550  WRITE   *DECLARES ARRAYS *A, *B, AND *C.
1551  AT      2412
1552  WRITE   *A              *B                    *C
1553  CALC    HEREY*40*HEREX*66*SISX*ROWS*5*SISY*COLS*1
1554  BACK    STRLEN,STRING,//////
1555  JOIN    DRAW
1556  JOIN    FILL
1557  CALC    HEREX*148*ROWS*2*COLS*2
1558  BACK    STRLEN,STRING,//////////////
1559  JOIN    DRAW
1560  JOIN    FILL
1561  CALC    HEREX*275*ROWS*2*COLS*3
```

```
1562  PACK    STRLEN,STRING,////////////////////
1563  JOIN    DRAW
1564  JOIN    FILL
1565  INHIBIT ERASE

                                                                                                    GAB
1566  UNIT    GAB
1567  BACK    ARRAYI/O
1568  AT      1405
1569  WRITE   *G*E*T *L*I*S*T(*A(1),*A(5))!
1570  AT      1430
1571  WRITE   *GETS 5 ITEMS AND STORES THEM
1572          IN THE CORRESPONDING ELEMENTS

 ------------------------------------------------------------------------------------------------------ ARRAY I/O
1574          OF ARRAY *A.
1575  FIGURE  504,508,608,604,504
1576  MODE    ERASE
1577  AT      2510
1578  WRITE   //////
1579  AT      2910
1580  WRITE   //////
1581  MODE    WRITE
1582  AT      2515
1583  WRITE   1
1584  AT      2915
1585  WRITE   2
1586  INHIBIT ERASE

                                                                                                    GAC
1587  UNIT    GAC
1588  BACK    ARRAYI/O
1589  AT      1805
1590  WRITE   *G*E*T *L*I*S*T(*B(*,1))!
1591  AT      1830
1592  WRITE   *GETS NEXT 4 ITEMS AND STORES
1593          THEM IN COLUMN 1 OF ARRAY *B.
1594  MODE    ERASE
1595  FIGURE  504,508,608,604,504
1596  CALC    HEREX=148*ROWS*4
1597  PACK    STRLEN,STRING,//////
1598  JOIN    FILL
1599  MODE    WRITE
1600  FIGURE  508,519,619,608,508
1601  AT      2525
1602  WRITE   3
1603          4
1604          5
1605          6
1606  INHIBIT ERASE

                                                                                                    GAD
1607  UNIT    GAD
1608  BACK    ARRAYI/O
1609  AT      2105
```

```
1610  WRITE   +G+E+T +L+I+S+T(+C)!
1611  AT      2130
1612  WRITE   +GETS NEXT 15 ITEMS AND STORES THEM
1613          IN ARRAY +C IN ROW MAJOR ORDER.
1614  FIGURE  543,520,620,643
1615  FIGURE  804,827,927,904
1616  MODE    ERASE
1617  FIGURE  508,519,619,608,508
1618  CALC    HEREX=262+STRLEN=13+ROWS=5
1619  PACK    STRING,//////////////////
1620  JOIN    FILL
1621  MODE    WRITE
1622  AT      2540
1623  WRITE    7     8     9
1624           10    11    12
1625           13    14    15
1626           16    17    18
1627           19    20    21
                                                                    68
1628  UNIT    GB
1629  BACK    GA
1630  ERASE
1631  AT      101
1632  WRITE   +L+I+S+T-+D+I+R+E+C+T+E+D +O+U+T+P+U+T
1633  AT      305
1634  WRITE   +IF ARRAY +A+A IS DECLARED AS +A+A(9,3) AND CONTAINS+!
1635  CALC    HEREY=280+HEREX=143+ROWS=9
1636  JOIN    DRAW
1637  PACK    STRLEN,STRING,+A+A( , )
1638  CALC    HEREX=96
1639  JOIN    FILL
1640  CALC    HEREX=296
1641  JOIN    FILL
1642  AT      416
1643  WRITE                 +A+A
1644           1 1     1     2     3    1 3
1645           2 1     4     5     6    2 3
1646           3 1     7     8     9    3 3
1647           4 1    10    11    12    4 3
1648           5 1    13    14    15    5 3
1649           6 1    16    17    18    6 3
1650           7 1    19    20    21    7 3
1651           8 1    22    23    24    8 3
1652           9 1    25    26    27    9 3
1653  DRAW    1405!1455!SKIP!1505!1555!SKIP!1425!3125
1654  AT      1505
1655  WRITE   +THE STATEMENT(S)+!    +WILL PRODUCE AS OUTPUT+!
1656          +P+U+T +L+I+S+T(+A+A(9,3))!   27
1657
1658          +P+U+T +L+I+S+T(+A+A(5,2))!   14
1659
1660          +P+U+T +L+I+S+T(+A+A(4,*))!   10 11 12
1661
1662          +P+U+T +L+I+S+T(+A+A(*,2))!   2  5  8 11 14 17 20 23 26
```

```
1663
1664          *P*U*T *L*I*S*T(*A*A)$          1  2  3  4  5  6  7  8  9  10
1665                                 11 12 13 14 15 16 17 18 19 20
1666                                 21 22 23 24 25 26 27
1667
1668          *D*O *I=1 *T*O 3$              1  4  7 10 13 16 19 22 25
1669           *P*U*T *S*K*I*P                 2  5  8 11 14 17 20 23 26
1670             *L*I*S*T(*A*A(*,*I))$         3  6  9 12 15 18 21 24 27
1671          *E*N*D$
                                                                                                    GC
1672  UNIT    GC
1673  BACK    GB
1674  ERASE
1675  AT      101
1676  WRITE   *D*A*T*A-*D*I*R*E*C*T*E*D *I*N*P*U*T
1677  AT      812
1678  WRITE   *IN *D*A*T*A-DIRECTED INPUT, THE FORM
 ---------------------------------------------------------------------------------------------------- ARRAY I/O
1680          OF THE *G*E*T STATEMENT IS*$
1681
1682            *G*E*T *D*A*T*A(LIST$$$$$*6*6*6*6,)$  OR  *G*E*T *D*A*T*A$
1683
1684          *WHEN LIST$$$$$*6*6*6*6 IS PRESENT, ONLY THOSE
1685          VARIABLES IN LIST$$$$$*6*6*6*6 CAN BE CHANGED.
1686          *ONLY ENTIRE ARRAY NAMES CAN APPEAR
1687          IN LIST$$$$$*6*6*6*6, NOT SUBSCRIPTED NAMES.
1688          *INPUT ITEMS SHOULD BE OF THE FORM*$
1689
1690            SUBSCRIPTED*6VARIABLE=DATA*6ITEM
1691
1692          AND MUST BE SEPARATED BY COMMAS OR
1693          BLANKS, AND MUST HAVE A SEMICOLON
1694          AFTER THE LAST DATA ITEM.
                                                                                                    GD
1695  UNIT    GD
1696  BACK    GC
1697  ERASE
1698  AT      101
1699  WRITE   *G*E*T *D*A*T*A(LIST$$$$$*6*6*6*6)$   *G*E*T *D*A*T*A$
1700  FIGURE  705,505,405,455
1701  FIGURE  1005,805,705,755
1702  AT      607
1703  WRITE   *A(3)=1$ *B(1,1)=2,*B(2,1)=3 *B(2,2,)=4,*B(1,2)=5$
1704
1705
1706          *C(2,1)=6,*C(2,2)=7  *C(2,3)=8$ *A(1)=9,*C(3,3)=10$
1707  AT      1105
1708  WRITE   *D*C*L(*A(3),*B(2,2),*C(3,3))$  *DECLARES ARRAYS *A, *B, AND *C.
1709  AT      2712
1710  WRITE   *A             *B                  *C
1711  CALC    HEREX*A*HEREY*61*ROWS*2*COLS*1
1712  BACK    STRLEN,STRING,//////
```

```
1713  JOIN     DRAW
1714  JOIN     FILL
1715  CALC     HEREX=150+ROWS*COLS*2
1716  PACK     STRLEN,STRING,//////////////
1717  JOIN     DRAW
1718  JOIN     FILL
1719  CALC     HEREX=277+ROWS*COLS*3
1720  PACK     STRLEN,STRING,////////////////////
1721  JOIN     DRAW
1722  JOIN     FILL
1723  INHIBIT ERASE

                                                                    GDA
1724  UNIT     GDA
1725  BACK     GC
1726  AT       1305
1727  WRITE    'G'E'T 'D'A'T'A('A');
1728  AT       1330
1729  WRITE    'GETS ITEMS UNTIL SEMICOLON, AND
1730           STORES THEM IN THE CORRESPONDING
1731           ELEMENTS OF ARRAY 'A.
1732  FIGURE   606,506,514,614,606
1733  AT       3010
1734  MODE     ERASE
1735  WRITE    //////
1736  MODE     WRITE
1737  AT       3015
1738  WRITE    1
1739  INHIBIT ERASE

                                                                    GDB
1740  UNIT     GDB
1741  BACK     GC
1742  AT       1615
1743  WRITE    'G'E'T 'D'A'T'A;
1744  AT       1630
1745  WRITE    'GETS ITEMS UNTIL SEMICOLON, AND
1746           STORES THEM IN THE CORRESPONDING
1747           ELEMENTS.
1748  AT       2820
1749  MODE     ERASE
1750  WRITE    //////////////
1751           //////////////
1752  FIGURE   606,506,514,614,606
1753  MODE     WRITE
1754  FIGURE   614,514,552,652,614
1755  AT       2825
1756  WRITE    2     5
1757           3     4
1758  INHIBIT ERASE

                                                                    GDC
1759  UNIT     GDC
1760  BACK     GC
1761  AT       1905
```

```
1762  WRITE   *G*E*T *D*A*T*A(*A,*C)$
1763  AT      1930
1764  WRITE   *GETS ITEMS UNTIL SEMICOLON, AND
1765          STORES THEM IN THE CORRESPONDING
1766          ELEMENTS OF ARRAY *A OR ARRAY *C.
1767  AT      2936
1768  MODE    ERASE
1769  WRITE   ////////////////////
1770  FIGURE  614,514,552,652,614
1771  MODE    WRITE
1772  FIGURE  906,806,835,935,906
1773  AT      2941
1774  WRITE   6     7     8
1775  INHIBIT ERASE

                                                                                                    GDD
1776  UNIT    GDD
1777  BACK    GC
1778  AT      2205
1779  WRITE   *G*E*T *D*A*T*A(*C)$
1780  AT      2230
1781  WRITE   *THIS IS AN ERROR. *THERE IS A DATA
1782          ITEM IN THE INPUT THAT DOES NOT
1783          APPEAR IN THE *D*A*T*A LIST. *AN ERROR

 ---------------------------------------------------------------------------------------------------------------------------------- ARRAY I/O
1785          MESSAGE IS PRINTED.
1786  AT      3048
1787  MODE    ERASE
1788  WRITE   //////
1789  FIGURE  906,806,835,935,906
1790  MODE    WRITE
1791  FIGURE  835,935,953,853,835
1792  AT      3052
1793  WRITE   10

                                                                                                    GE
1794  UNIT    GE
1795  BACK    GD
1796  ERASE
1797  AT      101
1798  WRITE   *D*A*T*A-*D*I*R*E*C*T*E*D *O*U*T*P*U*T
1799  AT      304
1800  WRITE   *D*A*T*A-DIRECTED OUTPUT IS THE SAME AS *L*I*S*T-DIRECTED
1801          OUTPUT, BUT THE VARIABLE NAMES ARE PLACED IN THE
1802          OUTPUT ALONG WITH THEIR CONTENTS. *FOR EACH *P*U*T *D*A*T*A
1803          STATEMENT, A SEMICOLON IS PRINTED AFTER THE LAST
1804          DATA ITEM.
1805
1806            *D*C*L *7(6,2) *I*N*I*T(1,2,3,4,5,6,7,8,9,10,11,12)$
1807  CALC    HEREY=144*HEREX=186*SISX=ROWS=4*COLS=2
1808  JOIN    DRAW
1809  PACK    STRLEN,STRING,*Z( , )
1810  CALC    HEREX=144
1811  JOIN    FILL
```

```
1812  CALC    HEREX*288
1813  JOIN    FILL
1814  AT      1021
1815  WRITE             *7
1816            1 1       1      2  1 2
1817            2 1       3      4  2 2
1818            3 1       5      6  3 2
1819            4 1       7      8  4 2
1820            5 1       9     10  5 2
1821            6 1      11     12  6 2
1822  DRAW    1705;1756;SKIP;1805;1856;SKIP;1725;3025
1823  AT      1805
1824  WRITE   *THIS STATEMENT*;        *WILL PRODUCE THIS OUTPUT*;
1825
1826          *P*U*T *D*A*T*A(*Z(1,1));      *Z(1,1)=1;
1827
1828          *P*U*T *D*A*T*A (*Z(5,*));     *Z(5,1)=9  *Z(5,2,)=10;
1829
1830          *P*U*T *D*A*T*A(*Z(*,2));      *Z(1,2)=2  *Z(2,2)=4  *Z(3,2)=6
1831                                  *Z(4,2)=8  *Z(5,2)=10 *Z(6,2)=12;
1832
1833          *P*U*T *D*A*T*A(*Z);           *Z(1,1)=1  *Z(1,2)=2  *Z(2,1)=3
1834                                  *Z(2,2)=4  *Z(3,1)=5  *Z(3,2)=6
1835                                  *Z(4,1)=7  *Z(4,2)=8  *Z(5,1)=9
1836                                  *Z(5,2)=10 *Z(6,1)=11 *Z(6,2)=12;
```

GF

```
1837  UNIT    GF
1838  BACK    GF
1839  ERASE
1840  AT      101
1841  WRITE   *E*D*I*T=*D*I*R*E*C*T*E*D *I*N*P*U*T
1842  AT      306
1843  WRITE   *IN *E*D*I*T-DIRECTED INPUT, THE INPUT ITEMS MUST BE
1844          FORMATED ACCORDING TO THE *G*E*T *E*D*I*T STATEMENT.
1845  AT      608
1846  WRITE   15   9 128   1 17  6 34820*H*E*L*L*O
1847  FIGURE  705,505,406,442
1848  INHIBIT ERASE
```

GFA

```
1849  UNIT    GFA
1850  BACK    GE
1851  AT      805
1852  WRITE   *D*C*L((*A(4)+*B(4,2))*F*I*X*E*D(3),
1853                *C(2) *C*H*A*R(5));
1854  AT      2413
1855  WRITE   *A              *B                  *C
1856  CALC    HEREY*24+HEREX*84+ROWS*1+SISX*STRLEN*2+COLS*1
1857  PACK    STRING,///
1858  JOIN    DRAW
1859  JOIN    FILL
1860  CALC    HEREX*206+COLS*2
1861  PACK    STRLEN,STRING,//////
1862  JOIN    DRAW
```

```
1863  JOIN    FILL
1864  CALC    HEREX*347*SISX*3*ROWS*1*COLS*1
1865  PACK    STRLEN,STRING,/////
1866  JOIN    DRAW
1867  JOIN    FILL
1868  INHIBIT ERASE

                                                                                GFB
1869  UNIT    GFB
1870  BACK    GE
1871  AT      1105
1872  WRITE   *G*E*T *E*D*I*T(*A)((4)(*X(1),*F(3)));
1873  AT      1133
1874  WRITE   *GETS FOUR ITEMS WITH FORMAT
1875          *F(3) SKIPPING ONE SPACE BEFORE
1876          EVERY ITEM, AND STORES THEM
1877          IN ARRAY *A.

------------------------------------------------------------------------------ ARRAY I/O
1879  CALC    HEREY*24*HEREX*84*ROWS*4
1880  PACK    STRLEN,STRING,///
1881  MODE    ERASE
1882  JOIN    FILL
1883  MODE    WRITE
1884  FIGURE  506,522,622,606,506
1885  AT      2512
1886  WRITE    15
1887            0
1888          128
1889            1
1890  INHIBIT ERASE

                                                                                GFC
1891  UNIT    GFC
1892  BACK    GE
1893  AT      1605
1894  WRITE   *G*E*T *E*D*I*T(*B(*,2))((4)*F(3));
1895  AT      1633
1896  WRITE   *GETS FOUR ITEMS WITH FORMAT
1897          *F(3), AND STORES THEM IN COLUMN
1898          2 OF ARRAY *B.
1899  CALC    HEREX*232
1900  MODE    ERASE
1901  FIGURE  506,522,622,606,506
1902  JOIN    FILL
1903  MODE    WRITE
1904  FIGURE  522,534,634,622,522
1905  AT      2530
1906  WRITE    17
1907             6
1908            34
1909          820
1910  INHIBIT ERASE

                                                                                GFD
```

```
1911  UNIT    GFD
1912  BACK    GF
1913  AT      2005
1914  WRITE   *G*F*T *E*D*I*T(*C(1))(*A(5));
1915  AT      2033
1916  WRITE   *GETS ONE ITEM WITH FORMAT
1917          *A(5), AND STORES IT IN *C(1).
1918  MOVE    ERASE
1919  FIGURE  522,534,634,622,522
1920  AT      2545
1921  WRITE   /////
1922  MOVE    WRITE
1923  FIGURE  534,539,639,634,534
1924  AT      2545
1925  WRITE   *H*E*L*L*O
```

GG

```
1926  UNIT    GG
1927  BACK    GF
1928  ERASE
1929  AT      101
1930  WRITE   *E*D*I*T-*D*I*R*E*C*T*E*D *O*U*T*P*U*T
1931  AT      410
1932  WRITE   *IN *E*D*I*T-DIRECTED OUTPUT, THE OUTPUT IS EDITED
1933          ACCORDING TO FORMATS<<<<<<*6*6*6*6*6*6 IN THE *P*U*T *E*D*I*T STATEMENT*;
1934  AT      715
1935  WRITE   *P*U*T *E*D*I*T (LIST<<<<*6*6*6*6) (FORMAT<<<<<<*6*6*6*6*6*6);
1936  AT      910
1937  WRITE   *IF ARRAY *F*L IS DECLARED AS
1938          *D*C*L *F*L(3,3) *E*I*X*F*D(6,2);
1939          AND CONTAINS*;
1940  CALC    HEREX*176+HEREX*106*SISX*3+ROWS*COLS*3
1941  JOIN    DRAW
1942  BACK    STRLEN,STRING,*F*L( , )
1943  CALC    HEREX*56
1944  JOIN    FILL
1945  CALC    HEREX*256
1946  JOIN    FILL
1947  AT      1311
1948  WRITE                 *F*L
1949          1 1 126.30045,72000.17   1 3
1950          2 1 007.00492.11001.00   2 3
1951          3 1 100.01369.68027.50   3 3
1952  DRAW    1805;1863;SKIP;1905;1963;SKIP;1825;2825
1953  AT      1905
1954  WRITE   *THIS STATEMENT*;        *WILL PRODUCE THIS OUTPUT*; (*6=BLANK)
1955
1956          *P*U*T *E*D*I*T(*F*L(3,2))      *6*63.6968*F*602
1957              (*F(12.4));
1958
1959          *P*U*T *E*D*I*T(*F*L(2,*))      *6*6*67.0*6492.1*6*6*61.0
1960              ((3)*F(6.1));
1961
1962          *P*U*T *E*D*I*T(*F*L)(*F(4));   *6126*6*646*6*6*60*6*6*67*6492*6*6*61*100*6371*6*628
```

```
                                                                                                                    GH
1963  UNIT    GH
1964  BACK    GG
1965  ERASE
1966  AT      101
1967  WRITE   *R*E*P*E*T*I*T*I*V*E *S*P*E*C*I*F*I*C*A*T*I*O*N
1968  AT      310
1969  WRITE   *WE CAN USE THE REPETITIVE SPECIFICATION
1970          TO INPUT AND OUTPUT ARRAYS.
1971          *THE FORM OF THE REPETITIVE SPECIFICATION IS*!
1972
1973             ( LIST<<<<<*^^6^6^6 *D*O SPECIFICATIONS<<<<<<<<<<<<<<<<*6*6*6*6*6*6*6*6*6*6*6*6*6*6 )
1974
1975          *EVERY SPECIFICATION NEEDS ITS OWN PAIR
1976          OF PARENTHESES. *FOR THE EXAMPLES, SUPPOSE
1977          THAT ARRAYS *A, *B, AND *C ARE DECLARED AS*!
1978
1979                *D*C*L(*A(15),*B(20,5),*C(5,10,15))!
1980  INHIBIT ERASE
```

```
                                                                                                                    GHA
1982  UNIT    GHA
1983  BACK    GG
1984  AT      1610
1985  WRITE         *G*E*T *L*I*S*T((*A(*I) *D*O *I=1 *T*O 15))!
1986          IS EQUIVALENT TO*!
1987                *D*O *I=1 *T*O 15!
1988                  *G*E*T *L*I*S*T(*A(*I))!
1989                *E*N*D!
1990          WHICH IS EQUIVALENT TO*!
1991                *G*E*T *L*I*S*T(*A)!
1992  INHIBIT ERASE

                                                                                                                    GHB
1993  UNIT    GHB
1994  BACK    GG
1995  AT      2510
1996  WRITE         *G*E*T *L*I*S*T((*B(*J,3) *D*O *J=1 *T*O 20))!
1997          IS EQUIVALENT TO*!
1998                *G*E*T *L*I*S*T(*B(*,3))!

                                                                                                                    GI
1999  UNIT    GI
2000  BACK    GH
2001  WRITE   *R*E*P*E*T*I*T*I*V*E *S*P*E*C*I*F*I*C*A*T*I*O*N   (LIST *D*O SPE
2002  WRITE   CIFICATIONS)
2003  AT      315
2004  WRITE   *D*C*L(*A(15),*B(20,5),*C(5,10,15))!
2005  AT      610
2006  WRITE   *P*U*T *L*I*S*T((*I,((*C(*I,*J,*K) *D*O *K=1 *T*O 15)
2007              *D*O *J=1 *T*O 10) *D*O *I=1 *T*O 5))!
2008  INHIBIT ERASE
```

GIA

```
2009  UNIT    GIA
2010  BACK    GH
2011  AT      1010
2012  WRITE   *THE ABOVE STATEMENT IS EQUIVALENT TO*;
2013
2014            *D*O *I=1 *T*O 5;
2015              *P*U*T *L*I*S*T(*I);
2016              *D*O *J=1 *T*O 10;
2017                *D*O *K=1 *T*O 15;
2018                *P*U*T *L*I*S*T(*C(*I,*J,*K));
2019                *E*N*D;
2020              *E*N*D;
2021            *E*N*D;
2022  INHERIT ERASE
```

GIB

```
2023  UNIT    GIB
2024  BACK    GH
2025  AT      2210
2026  WRITE   WHICH IN TURN IS EQUIVALENT TO*;
2027
2028            *D*O *I=1 *T*O 5;
2029              *P*U*T *L*I*S*T(*I,*C(*I,*,*));
2030            *E*N*D;
2031  NEXT    ENDTOPIC
```

MATRMULT

```
2032  UNIT    MATRMULT
2033  BACK    CONTENTS
2034  ERASE
2035  AT      101
2036  WRITE   *M*A*T*R*I*X *M*U*L*T*I*P*L*I*C*A*T*I*O*N
2037  AT      410
2038  WRITE   *IF ARRAYS *A, *B, AND *C ARE DECLARED AS FOLLOWS*;
2039
2040             *D*E*C*L*A*R*E (*A(*L,*N),*B(*L,*M),*C(*M,*N)) *F*I*X*E*D;
2041
2042          *THEN, *MATRIX *MULTIPLICATION IS DEFINED AS*;
2043
2044                        *M
2045            *A(*I,*J) = ΣHI> *B(*I,*K)*C(*K,*J)
2046                        *K=1
2047  INHERIT ERASE
```

HA

```
2048  UNIT    HA
2049  BACK    CONTENTS
2050  AT      1510
2051  WRITE   *WE CAN COMPUTE *A(*I,*J) IN ANY OF THE
2052          FOLLOWING WAYS*;
2053
2054          *WITH A *D*O LOOP*;
2055            *A=0;
2056            ....
```

```
2057              *D*O *K=1 *T*O *M;
2058                *A(*I,*J) = *A(*I,*J) + *B(*I,*K)**C(*K,*J);
2059              *E*N*D;
2060    INHIBIT ERASE
```

4AA

```
2061    UNIT    4AA
2062    BACK    CONTENTS
2063    AT      2610
2064    WRITE   *OR WITH A SINGLE STATEMENT*;
2065
2066              *A(*I,*J) = *S*U*M(*B(*I,*)**C(*,*J));
```

4B

```
2067    UNIT    4B
2068    BACK    MATRMULT
2069    ERASE
2070    AT      205
2071    WRITE   *THIS IS A PROGRAM TO PERFORM *MATRIX *MULTIPLICATION
2072            ON TWO ARRAYS GIVEN AS INPUT*;
2073    AT      715
2074    WRITE   *M*A*T*M*U*L*T*; *P*R*O*C*E*D*U*R*E *O*P*T*I*O*N*S(*M*A*I*N);
2075              *D*C*L  (*A(4,5),*B(4,3),*C(3,5)) *F*I*X*E*D;
2076              *G*E*T *L*I*S*T(*B,*C);
2077              *D*O *I=1 *T*O 4;
2078                *D*O *J=1 *T*O 5;
2079                *A(*I,*J)=*S*U*M(*B(*I,*)**C(*,*J));
2080                *E*N*D;
2081              *E*N*D;
```

---------- MATMUL-CON

```
2083              *P*U*T *D*A*T*A(*A,*B,*C);
2084              *E*N*D *M*A*T*M*U*L*T;
2085    INHIBIT ERASE
```

4BA

```
2086    UNIT    4BA
2087    BACK    MATRMULT
2088    FIGURE  714,750,850,814,716
2089    AT      2010
2090    WRITE   *DECLARES ARRAYS *A, *B, AND *C.
2091    INHIBIT ERASE
```

4BB

```
2092    UNIT    4BB
2093    BACK    MATRMULT
2094    MODE    ERASE
2095    FIGURE  714,750,850,816,716
2096    MODE    REWRITE
2097    FIGURE  816,831,931,916,816
2098    AT      2010
2099    WRITE   *GETS ARRAYS *B AND *C FROM *INPUT.
2100    INHIBIT ERASE
```

4BC

```
2101  UNIT    HBC
2102  BACK    MATRMULT
2103  MODE    ERASE
2104  FIGURE  816,831,931,916,816
2105  MODE    REWRITE
2106  FIGURE  1118,1145,1245,1218,1118
2107  AT      2010
2108  WRITE   *COMPUTES *A(*I,*J) ACCORDING TO DEFINITION
2109          OF *MATRIX *MULTIPLICATION*!
2110                      *M
2111             *A(*I,*J) = ≥HI≥ *B(*I,*K)**C(*K,*J)
2112                      *K=1
2113  INHIBIT ERASE
```

HBD

```
2114  UNIT    HBD
2115  BACK    MATRMULT
2116  MODE    ERASE
2117  FIGURE  1118,1145,1245,1218,1118
2118  MODE    REWRITE
2119  FIGURE  916,945,1445,1416,916
2120  AT      2010
2121  WRITE   *COMPUTES *A(*I,*J) FOR ALL VALUES OF *I AND *J.
2122          I.E. FOR THE ENTIRE ARRAY.
2123  INHIBIT ERASE
```

HBE

```
2124  UNIT    HBE
2125  BACK    MATRMULT
2126  LAB     HB
2127  MODE    ERASE
2128  FIGURE  916,945,1445,1416,916
2129  MODE    REWRITE
2130  FIGURE  1416,1433,1533,1516,1416
2131  AT      2010
2132  WRITE   *OUTPUTS ARRAYS *A, *B, AND *C .
2133  ERASE   55
2134  AT      3225
2135  WRITE   PRESS *L*A*B TO GO OVER THE PROGRAM AGAIN
2136  NEXT    ENDTOPIC
```

CONNMATR

```
2137  UNIT    CONNMATR
2138  BACK    CONTENTS
2139  ERASE
2140  AT      101
2141  WRITE   *C*O*N*N*E*C*T*I*V*I*T*Y *M*A*T*R*I*X
2142  AT      405
2143  WRITE   *LET*7S SUPPOSE WE HAVE A MAP OF THE ROAD SYSTEM
2144          BETWEEN SIX CITIES*!
2145  JOIN    CITY
2146  AT      825
2147  WRITE   *HOW DO WE KNOW IF THERE IS A WAY
2148          OF GOING FROM CITY 1 TO CITY 5*/
2149          *WE WOULD LOOK FIRST FOR A DIRECT
```

```
2150          ROAD AND THEN FOR AN INDIRECT ONE.
2151          *WE CANNOT GO FROM 1 TO 5 DIRECTLY.
2152          BUT WE CAN GO FROM 1 TO 2 AND FROM
2153          THERE TO 5 (THEY ARE CONNECTED).
2154  AT      1610
2155  WRITE   *CAN WE GO FROM 1 TO 4*/ *THERE IS NOT A DIRECT
2156          ROAD. *FROM 1 WE CAN GO TO 2 ONLY; FROM 2 WE
2157          CAN GO BACK TO 1 OR GO TO 5; FROM 5 WE CAN
2158          GO BACK TO 2 OR GO TO 6; FROM 6 WE CAN ONLY
2159          GO BACK TO 5. *THEREFORE, WE CANNOT GO FROM
2160          CITY 1 TO CITY 4 (THEY ARE NOT CONNECTED).
2161  INHIBIT ERASE

                                                                                T
2162  UNIT    I
2163  BACK    CONTENTS
2164  AT      2410
2165  WRITE   *A *CONNECTIVITY *MATRIX IS A MATRIX THAT TELLS
2166          US WHICH CITIES ARE CONNECTED.
2167          *WE OBTAIN THE *CONNECTIVITY *MATRIX FROM THE
2168          *ADJACENCY *MATRIX (PRESS *N*E*X*T).

                                                                                TA
2169  UNIT    IA
2170  BACK    CONNMATR
2171  ERASE
2172  AT      208
2173  WRITE   *WE CAN STORE THE ROAD SYSTEM IN A SQUARE MATRIX *A
2174          CALLED THE *ADJACENCY *MATRIX AS FOLLOWS*!
2175          WE STORE A 1 IN *A(*I,*J) IF THERE IS A DIRECT ROAD
2176          FROM CITY *I TO CITY *J, AND A 0 OTHERWISE.
2177  JOIN    CITY

  ------------------------------------------------------------------------------------------------------------------------------ CONNMATR
2179  CALC    HEREY*80*HEREX*256*ROWS*COLS*4*SISX*1*SISY*1
2180  JOIN    DRAW
2181  AT      2304
2182  WRITE         *A
2183            1 2 3 4 5 6
2184          1 1 1 0 0 0 0
2185          2 1 1 0 0 1 0
2186          3 0 0 1 1 0 0
2187          4 0 0 1 1 0 0
2188          5 0 1 0 0 1 1
2189          6 0 0 0 0 1 1
2190          *ADJACENCY
2191                *MATRIX
2192  INHIBIT ERASE

                                                                                TAA
2193  UNIT    IAA
2194  BACK    CONNMATR
2195  AT      726
2196  WRITE   *WE DEFINE OPERATION *AA2 = *A ≥*≥ *A
2197          AS FOLLOWS*!
```

```
2198            *A^2(*I,*J) = ≥AR≥ (*A(*I,*K) ¬* *A(*K,*J))
2199                          *K
2200          WHERE ≥AR≥ IS THE BIT *O*R OPERATION.
2201          AND ¬* IS THE BIT *A*N*D OPERATION.
2202          *A(*I,*J) WILL CONTAIN A 1 ONLY IF
2203          *A(*I,*K) AND *A(*K,*J) CONTAIN A 1 FOR
2204          SOME *K.
2205  INHIBIT ERASE
                                                                        !AB
2206  UNIT    IAB
2207  BACK    CONNMATR
2208  AT      17,8
2209  WRITE   *A^2(*I,*J) WILL CONTAIN A 1 IF CITIES *I AND *J ARE
2210          CONNECTED BY AT MOST TWO ROADS.
2211  CALC    HEREX=152
2212  JOIN    DRAW
2213  AT      2319
2214  WRITE           *A^2
2215            1 2 3 4 5 6
2216          1 1 1 0 0 1 0
2217          2 1 1 0 0 1 1
2218          3 0 0 1 1 0 0
2219          4 0 0 1 1 0 0
2220          5 1 1 0 0 1 1
2221          6 0 1 0 0 1 1
2222  INHIBIT ERASE
                                                                        !AC
2223  UNIT    IAC
2224  BACK    CONNMATR
2225  AT      17,8
2226  ERASE   11,
2227  AT      17,8
2228  WRITE   *A^3(*I,*J) WILL CONTAIN A 1 IF CITIES *I AND *J ARE
2229          CONNECTED BY AT MOST THREE ROADS.
2230  CALC    HEREX=272
2231  JOIN    DRAW
2232  AT      2334
2233  WRITE           *A^3
2234            1 2 3 4 5 6
2235          1 1 1 0 0 1 1
2236          2 1 1 0 0 1 1
2237          3 0 0 1 1 0 0
2238          4 0 0 1 1 0 0
2239          5 1 1 0 0 1 1
2240          6 1 1 0 0 1 1
2241  INHIBIT ERASE
                                                                        !AD
2242  UNIT    IAD
2243  BACK    CONNMATR
2244  AT      19,8
2245  WRITE   *A^4(*I,*J) WILL CONTAIN A 1 IF CITIES *I AND *J ARE
2246          CONNECTED BY AT MOST FOUR ROADS (*A^4=*A^3≥*≥*A).
```

```
2247            ↑AND SO ON (↑A↓5=↑A↓4≥↓≥↑A).
2248    CALC    HERE¥←392
2249    JOIN    DRAW
2250    AT      2349
2251    WRITE           ↑A↓5
2252              1 2 3 4 5 6
2253            1 1 1 0 0 1 1
2254            2 1 1 0 0 1 1
2255            3 0 0 1 1 0 0
2256            4 0 0 1 1 0 0
2257            5 1 1 0 0 1 1
2258            6 1 1 0 0 1 1
2259    INHIBIT ERASE
```

↑AE

```
2260    UNIT    IAE
2261    BACK    CONNMATR
2262    AT      1708
2263    ERASE   250
2264    AT      1708
2265    WRITE   ↑A↓5 IS THE ↑CONNECTIVITY ↑MATRIX ↑C. BECAUSE WE
2266            CAN TRAVEL FROM ONE CITY TO ANOTHER USING AT THE
2267            MOST 5 ROADS. IF THE CITIES ARE CONNECTED.
2268            ↑IF THERE ARE N CITIES. ↑A↓N↓A=↓1 WILL BE THE
2269            ↑CONNECTIVITY ↑MATRIX (↑C=↑A↓N↓A=↓1).
2270    AT      2354
2271    MODE    REWRITE
2272    WRITE   ↑C=↑A↓5
2273    MODE    WRITE
2274    AT      3143
2275    WRITE   ↑CONNECTIVITY
2276              ↑MATRIX
```

↑8

```
2277    UNIT    IH
2278    BACK    IA
2279    ERASE
2280    AT      410
2281    WRITE   ↑THERE IS A BETTER WAY OF GETTING THE
2282            ↑CONNECTIVITY ↑MATRIX. ↑FOR 9 CITIES FOR
2283            EXAMPLE. ↑C = ↑A↓8.  ↑WE OBTAIN ↑A↓8 IN ANY
```



```
2285            OF THESE WAYS↑!
2286    AT      918
2287    WRITE   ↑A↑2=↑A>↑≥↑A      ↑A↑2=↑A>↑≥↑A
2288            ↑A↓3=↑A↓2≥↓≥↑A     ↑A↓4=↑A↓2≥↓≥↑A↑2
2289            ↑A↓4=↑A↓3≥↓≥↑A     ↑A↓8=↑A↓4≥↓≥↑A↓4
2290            ↑A↓5=↑A↓4≥↓≥↑A
2291            ↑A↓6=↑A↓5≥↓≥↑A
2292            ↑A↓7=↑A↓6≥↓≥↑A
2293            ↑A↓8=↑A↑7≥↓≥↑A
2294    AT      1710
2295    WRITE   ↑IF WE NEED TO DO M OPERATIONS IN THE FIRST
2296            CASE. IN THE SECOND CASE WE NEED TO DO ONLY
```

```
2297          LOGV2M OPERATIONS.
2298          ↑TO OBTAIN THE ↑CONNECTIVITY ↑MATRIX ↑C FOR N
2299          CITIES, WE NEED TO APPLY THE ≥↑≥ OPERATOR
2300          N-1 TIMES IN THE FIRST CASE. AND ONLY
2301          [LOGV2N] TIMES IN THE SECOND CASE (ROUND
2302          LOGV2N TO THE NEXT INTEGER).
2303
2304          ↑IF TWO CITIES ARE NOT CONNECTED BY AT MOST
2305          N-1 ROADS, THEY ARE NOT CONNECTED. ↑THUS,
2306          ↑C = ↑AAN-A1 = ↑AAN = ↑AANA↑A1 ...ETC.

                                                                                ↑C
2307  UNIT    IC
2308  BACK    IB
2309  ERASE
2310  AT      205
2311  WRITE   ↑THIS IS A PROGRAM TO OBTAIN THE ↑CONNECTIVITY
2312          ↑MATRIX FOR THE PROBLEM OF THE SIX CITIES↑!
2313  AT      505
2314  WRITE   ↑C↑O↑N↑M↑A↑T↑R↑!  ↑P↑R↑O↑C↑E↑D↑U↑R↑E ↑O↑P↑T↑I↑O↑N↑S(↑M↑A↑I↑N)!
2315  AT      615
2316  WRITE   ↑D↑C↑L(↑A(6,6),↑C(6,6)) ↑B↑I↑T(1)!
2317          ↑G↑E↑T ↑L↑I↑S↑T(↑A)!
2318          ↑P↑U↑T ↑L↑I↑S↑T(↑7↑A↑D↑J↑A↑C↑E↑N↑C↑Y ↑M↑A↑T↑R↑I↑X↑7)!
2319          ↑D↑O ↑I=1 ↑T↑O 6!
2320            ↑P↑U↑T ↑S↑K↑I↑P ↑L↑I↑S↑T(↑A(↑I,*))!
2321          ↑E↑N↑D!
2322          ↑D↑O ↑N=1 ↑T↑O 3!
2323            ↑D↑O ↑I=1 ↑T↑O 6!
2324              ↑D↑O ↑J=1 ↑T↑O 6!
2325              ↑C(↑I,↑J)=↑A↑N↑Y(↑A(↑I,*) -↑ ↑A(*,↑J))!
2326              ↑E↑N↑D!
2327            ↑E↑N↑D!
2328            ↑A=↑C!
2329          ↑E↑N↑D!
2330          ↑P↑U↑T ↑S↑K↑I↑P(2) ↑L↑I↑S↑T(↑7↑C↑O↑N↑N↑E↑C↑T↑I↑V↑I↑T↑Y ↑M↑A↑T↑R↑I↑X↑7)!
2331          ↑D↑O ↑I=1 ↑T↑O 6!
2332            ↑P↑U↑T ↑S↑K↑I↑P ↑L↑I↑S↑T(↑A(↑I,*))!
2333          ↑E↑N↑D!
2334          ↑E↑N↑D ↑C↑O↑N↑M↑A↑T↑R!
2335  INHIBIT ERASE

                                                                                ↑CA
2336  UNIT    ICA
2337  BACK    IH
2338  FIGURE  614,644,1144,1114,614
2339  AT      2705
2340  WRITE   ↑INPUTS THE ↑ADJACENCY ↑MATRIX, AND OUTPUTS IT
2341          WITH A TITLE.
2342  INHIBIT ERASE

                                                                                ↑CB
2343  UNIT    ICB
2344  BACK    IB
2345  MODE    ERASE
```

```
2346  FIGURE  614,644,1144,1114,614
2347  MODE    REWRITE
2348  FIGURE  1418,1447,1547,1518,1418
2349  AT      2705
2350  WRITE   *COMPUTES *C(*I,*J) ACCORDING TO THE DEFINITION*:
2351
2352
2353            *A**2(*I,*J) = ≥AR≥ (*A(*I,*K)¬*A(*K,*J))
2354                          *K
2355  INHIBIT ERASE
```

1CC

```
2356  UNIT    ICC
2357  BACK    IB
2358  MODE    ERASE
2359  FIGURE  1418,1447,1547,1518,1418
2360  MODE    REWRITE
2361  FIGURE  1216,1247,1747,1716,1216
2362  AT      2705
2363  WRITE   *COMPUTES *C(*I,*J) FOR ALL *I,*J (*C = *A ≥*¬ ≥*A).
2364  INHIBIT ERASE
```

1CD

```
2365  UNIT    ICD
2366  BACK    IB
2367  MODE    ERASE
2368  FIGURE  1216,1247,1747,1716,1216
2369  MODE    REWRITE
2370  FIGURE  1716,1721,1821,1816,1716
2371  AT      2705
2372  WRITE   *ASSIGNS *C TO *A (*EQUIVALENT TO *A = *C = *A ≥*≥ *A).
2373  INHIBIT ERASE
```

1CE

```
2374  UNIT    ICE
2375  BACK    IB
2376  MODE    ERASE
2377  FIGURE  1716,1721,1821,1816,1716
```

```
2379  MODE    REWRITE
2380  FIGURE  1114,1147,1947,1914,1114
2381  AT      2705
2382  WRITE   *DO LOOP TO COMPUTE *A = *A ≥*≥ *A 3 TIMES. *C = *A ≤^B
2383          WHEN THE PROGRAM GOES OUT OF THE LOOP.
2384  INHIBIT ERASE
```

1CF

```
2385  UNIT    ICF
2386  BACK    IB
2387  LAB     IC
2388  MODE    ERASE
2389  FIGURE  1114,1147,1947,1914,1114
2390  MODE    REWRITE
2391  FIGURE  1914,1955,2355,2314,1914
```

```
2392  AT      2705
2393  WRITE   *PRINTS OUT THE *CONNECTIVITY *MATRIX WITH ITS TITLE.
2394  ERASE   50
2395  AT      3225
2396  WRITE   PRESS *L*A*B TO GO OVER THE PROGRAM AGAIN
2397  NEXT    ENDTOPIC

                                                                                DISTMATR
2398  UNIT    DISTMATR
2399  BACK    CONTENTS
2400  ERASE
2401  AT      101
2402  WRITE   *D*I*S*T*A*N*C*E *M*A*T*R*I*X
2403  AT      205
2404  WRITE   *SUPPOSE WE HAVE A MAP SHOWING THE ROADS AND THE
2405          DISTANCES BETWEEN 6 CITIES.
2406  JOIN    CITY
2407  LINE    1008,810
2408  LINE    1311,1317
2409  AT      813
2410  WRITE   55
2411  AT      1314
2412  WRITE   27
2413  AT      48,362
2414  WRITE   33
2415                67
2416  AT      48,314
2417  WRITE   19          40
2418  INHIBIT ERASE

                                                                                I
2419  UNIT    J
2420  BACK    CONTENTS
2421  AT      405
2422  WRITE   *NOW, LETS STORE THE DISTANCES IN A SQUARE MATRIX *A CALLED
2423          THE *DIRECT *DISTANCE *MATRIX. *STORE 999 OR THE HIGHEST
2424          FIGURE POSSIBLE IN *A(*I,*J). IF THERE IS NOT A DIRECT
2425                              ROAD FROM CITY *I TO CITY *J.
2426  CALC    HEREY,40*HEREX,66*ROWS,COLS,3*SISX,2*SISY,1
2427  JOIN    DRAW
2428  AT      2309
2429  WRITE                 *A
2430            1  2  3  4  5  6
2431          1  0 55999999999 33
2432          2 55  0999999 67999
2433          3999999  0 40999999
2434          4999999 40  0 27999
2435          5999 67999 27  0 19
2436          6 33999999999 19  0
2437          *DIRECT *DISTANCE *MATRIX
2438  INHIBIT ERASE

                                                                                JA
2439  UNIT    JA
2440  BACK    CONTENTS
```

```
2441  AT      925
2442  WRITE   †WE DEFINE †AA2 = †A ≥#≥ †A AS FOLLOWS†!
2443
2444          †AA2(†I,†J) = MIN (†A(†I,†K)+†A(†K,†J))
2445                        †K
2446  AT      1325
2447  WRITE   †AA2(†I,†J) WILL CONTAIN THE SHORTEST
2448          DISTANCE FROM CITY †I TO CITY †J USING
2449          AT MOST 2 ROADS†! †A(†I,†K) AND †A(†K,†J).
2450  INHIBIT ERASE
```

JAB

```
2451  UNIT    JAB
2452  BACK    CONTENTS
2453  AT      1605
2454  WRITE   †AA3(†I,†J) WILL CONTAIN THE SHORTEST DISTANCE FROM CITY †I
2455          TO CITY †J USING AT MOST 3 ROADS (†AA3 = †AA2≥ ≠≥ †A).
2456          †AND SO ON.
2457  INHIBIT ERASE
```

JAC

```
2458  UNIT    JAC
2459  BACK    CONTENTS
2460  AT      2005
2461  WRITE   †WE OBTAIN THE †DISTANCE †MATRIX †D FROM MATRIX †A WITH
2462          THE ≥#≥ OPERATOR. †FOR N CITIES, †D = †AANA-1.
2463  CALC    HERE×+296
2464  JOIN    DRAW
2465  AT      2337
2466  WRITE             †D=†AA5
2467                1  2  3  4  5  6
2468          1  0 55 19 79 52 33
2469          2 55  0 34 94 67 86
2470          3 19 34  0 40 67 86
2471          4 79 94 40  0 27 46
2472          5 52 67 67 27  0 19
2473          6 33 86 86 46 19  0
2474            †DISTANCE †MATRIX
2475  INHIBIT ERASE
```

JB

```
2476  UNIT    JB
2477  BACK    DISTMATR
2478  ERASE
2479  AT      1010
2480  WRITE   †THE SHORTEST DISTANCE BETWEEN TWO CITIES
2481          WILL BE ATTAINED WHEN WE NEVER VISIT ONE
```

DISTMATR

```
2483          CITY MORE THAN ONCE, OTHERWISE WE ARE
2484          LOOPING.
2485          †THEN, †AANA-1 WILL BE THE †DISTANCE †MATRIX †D
2486          FOR N CITIES, I.E. †D = †AANA-1.
2487
2488          †ALSO, †AAN, †AANA+1, ETC. WILL BE EQUAL TO †AANA-1
```

```
2489            SINCE WE WILL NEVER GET A SHORTER DISTANCE
2490            THAN THE ONE CONTAINED IN ↑AANA=A1, ↑IN OTHER
2491            WORDS, ↑D = ↑AANA=A1 = ↑AAN = ↑AANA↑A1 = ↑AANA↑A2...

                                                                        JC
2492    UNIT    JC
2493    BACK    JB
2494    ERASE
2495    AT      205
2496    WRITE   ↑THIS IS A PROGRAM TO OBTAIN THE ↑DISTANCE ↑MATRIX
2497            USING THE OPERATOR 2↑2 AS DEFINED↑;
2498
2499            ↑A↑2(↑I,↑J) = MIN (↑A(↑I,↑K)↑↑A(↑K,↑J))
2500                          ↑K
2501    AT      810
2502    WRITE   ↑D↑I↑S↑T↑; ↑P↑R↑O↑C ↑O↑P↑T↑I↑O↑N↑S(↑M↑A↑I↑N);
2503    AT      915
2504    WRITE   ↑D↑C↑L (↑A(6,6),↑D(6,6)) ↑F↑I↑X↑E↑D;
2505            ↑G↑E↑T ↑L↑I↑S↑T(↑A);
2506            ↑P↑U↑T ↑D↑A↑T↑A(↑A);
2507            ↑D=↑A;
2508            ↑D↑O ↑N=1 ↑T↑O 3;
2509             ↑D↑O ↑I=1 ↑T↑O 5;
2510              ↑D↑O ↑J=↑I↑1 ↑T↑O 6;
2511               ↑D↑O ↑K=1 ↑T↑O 6;
2512               ↑T↑F ↑D(↑I,↑J)>↑A(↑I,↑K)↑↑A(↑K,↑J) ↑T↑H↑E↑N
2513                  ↑D(↑I,↑J)=↑D(↑J,↑I)=↑A(↑I,↑K)↑↑A(↑K,↑J);
2514               ↑E↑N↑D;
2515              ↑E↑N↑D;
2516             ↑E↑N↑D;
2517             ↑A=↑D;
2518            ↑E↑N↑D;
2519            ↑P↑U↑T ↑D↑A↑T↑A(↑D);
2520            ↑E↑N↑D ↑D↑I↑S↑T;
2521    INHIBIT ERASE

                                                                        JCA
2522    UNIT    JCA
2523    BACK    JB
2524    FIGURE  914,927,1127,1114,914
2525    AT      2705
2526    WRITE   ↑READS IN AND PRINTS OUT THE ↑DIRECT ↑DISTANCE ↑MATRIX.
2527    INHIBIT ERASE

                                                                        JCB
2528    UNIT    JCB
2529    BACK    JB
2530    MODE    ERASE
2531    FIGURE  914,927,1127,1114,914
2532    MODE    REWRITE
2533    FIGURE  1114,1119,1219,1214,1114
2534    AT      2705
2535    WRITE   ↑INITIALIZES ↑D TO ↑A FOR THE FIRST COMPARISONS.
2536    INHIBIT ERASE
```

```
                                                                                                                    JCC
2537  UNIT    JCC
2538  BACK    JB
2539  MODE    ERASE
2540  FIGURE  1114,1119,1219,1214,1114
2541  MODE    REWRITE
2542  FIGURE  1617,1650,1850,1817,1617
2543  AT      2705
2544  WRITE   *CHANGES *D(*I,*J) AND *D(*J,*I) IF LARGER THAN THE SUM
2545          *A(*I,*K)+*A(*K,*J), ACCORDING TO THE ABOVE FORMULA.
2546  INHIBIT ERASE

                                                                                                                    JCD
2547  UNIT    JCD
2548  BACK    JB
2549  MODE    ERASE
2550  FIGURE  1617,1650,1850,1817,1617
2551  MODE    REWRITE
2552  FIGURE  1315,1350,2150,2115,1315
2553  AT      2705
2554  WRITE   *OBTAINS *D(*I,*J) FOR ALL *I, *J. *SINCE THE ARRAY IS
2555          SYMMETRICAL, CALCULATIONS ARE DONE IN ONE HALF
2556          OF THE ARRAY ONLY.
2557  INHIBIT ERASE

                                                                                                                    JCE
2558  UNIT    JCE
2559  BACK    JB
2560  MODE    ERASE
2561  FIGURE  1315,1350,2150,2115,1315
2562  MODE    REWRITE
2563  FIGURE  2115,2120,2220,2215,2115
2564  AT      2705
2565  WRITE   *SETS *A = *D = *A ≥*≥ *A.
2566  ERASE   120
2567  INHIBIT ERASE

                                                                                                                    JCF
2568  UNIT    JCF
2569  BACK    JB
2570  MODE    ERASE
2571  FIGURE  2115,2120,2220,2215,2115
2572  MODE    REWRITE
2573  FIGURE  1214,1250,2350,2314,1214
2574  AT      2705
2575  WRITE   *PERFORMS *A = *D = *A≥ *≥ *A 3 TIMES, GIVING AT THE END
2576  AT      2805
2577  MODE    WRITE
2578  WRITE   *A = *D = *A^8, *SINCE THERE ARE 6 CITIES, *D = *A^5 = *A^8.
2579  INHIBIT ERASE

                                                                                                                    JCG
2580  UNIT    JCG
```

```
2582  BACK    JB
2583  LAB     JC
2584  MODE    ERASE
2585  FIGURE  1214,1250,2350,2314,1214
2586  MODE    REWRITE
2587  FIGURE  2314,2327,2427,2414,2314
2588  AT      2705
2589  WRITE   *PRINTS OUT THE *DISTANCE *MATRIX.
2590  ERASE   100
2591  AT      3225
2592  WRITE   PRESS *L*A*B TO GO OVER THE PROGRAM AGAIN

                                                                                ENDTOPIC
2593  UNIT    ENDTOPIC
2594  LAB     N29=2,DECLARAT.CONTENTS
2595  BACK    N29=2,AD,T2,EE,FD,GI,HB,IC,JC
2596  AT      1210
2597  WRITE   *THIS IS THE END OF THE TOPIC. *PRESS
2598            *R*A*C*K TO GO TO THE LAST FRAME,
2599            *L*A*B TO GO TO THE CONTENTS PAGE, OR
2600            *N*E*X*T TO STUDY NEXT TOPIC.

                                                                                NEWTOPIC
2601  UNIT    NEWTOPIC
2602  *
2603  CALCC   N29=2,N30*N30+1,N29*N29,1*N30*0
2604  JUMP    N29=3,X,ENTIRE,PARTS,ARRAYI/O,MATRMULT,CONNMATR,DISTMATR,CONTENTS
2605  JUMP    N30=1,LINEAR,MULTIDIM,DRILL1,INITIAL,DRILL2

                                                                                T2
2606  UNIT    T2
2607  JUMP    N30=1,BF,CE,X,DH

                                                                                ENDLESS
2608  UNIT    ENDLESS
2609  BASE
2610  END     LESSON

                                                                                DRAW
2611  UNIT    DRAW
2612  CALC    SISX*8*SISX*COLS
2613          SISY*16*SISY*ROWS
2614  ORIGIN  HEREX-1,HEREY-SISY,16
2615  AXES    SISX,SISY
2616  SCALEX  COLS
2617  SCALEY  ROWS
2618  MARKX   1,1,1
2619  MARKY   1,1,1
2620  CALC    SISX,SISX/8/COLS
2621          SISY,SISY/16/ROWS

                                                                                FILL
2622  UNIT    FILL
2623  ZERO    COUNT
```

```
2624  ENTRY   IN
2625  AT      HEREX,HEREY
2626  SHOWA   STRING,STRLEN
2627  CALC    HEREY=HEREY-16
2628          COUNT=COUNT+1
2629  GOTO    COUNT=ROWS,IN,X
2630  CALC    HEREY=HEREY+16*ROWS

                                                                    CITY
2631  UNIT    CITY
2632  AT      810
2633  WRITE   >C>       >C
2634  AT      1107
2635  WRITE   >C>             >C
2636  AT      1410
2637  WRITE   >C>       >C
2638  FIGURE  811,817,1311,1107
2639  LINE    1120,1318
2640  MODE    ERASE
2641  AT      810
2642  WRITE   1      2
2643  AT      1107
2644  WRITE   6            3
2645  AT      1410
2646  WRITE   5      4
2647  MODE    WRITE
```

LESSON PL1ARRAY AT 17.21.26. ON 06/13/73

UNIT	BLOCK	UNIT LOCATION	REFERENCES TO UNIT						
AA	INDEX-BASI	124	190	213					
AAA	INDEX-BASI	156							
AAB	BASICS	181							
AB	BASICS	189	230						
ABA	BASICS	212							
AC	BASICS	229	243						
AD	BASICS	242	2595						
ARRAYI/O	PARTS-I/O	1511	101	1530	1546	1547	1588	1608	2604
B	DCL-LINEAR	302	495						
BA	DCL-LINEAR	319	353						
BAA	LINEAR	337							
BASICS	INDEX-BASI	103	101	125	157	182			
BB	LINEAR	352	381						
BC	LINEAR	380	401						
BD	LINEAR	400	427	452					
BE	LIN-MULTID	426	2607						
BEA	LIN-MULTID	451							
BOUND	DR-INITIAL	710	738						
BOUND1	DR-INITIAL	717	713	716					
BOUND2	DR-INITIAL	739	719						
BOUND3	DR-INITIAL	748	738						
C	MULTIDIMEN	527	492	502					
CA	MULTIDIMEN	535	550	560					
CB	MULTIDIMEN	549	573						
CBA	MULTIDIMEN	559							
CC	MULTIDIMEN	572	598						
CD	MULTIDIMEN	597	616						
CE	MUL-DRILL1	615	2607						
CHGF	MUL-DRILL1	660	675						
CITY	-SUBROUTIN	2631	2145	2177	2406				
CONNMATR	MATMUL-CON	2137	102	2170	2194	2217	2224	2243	2261
			2604						
CONTENTS	INDEX-BASI	64	104	260	279	1122	1384	1402	1512
			2033	2049	2062	2138	2163	2399	2420
			2440	2452	2459	2594	2604		
DA	DR-INITIAL	778	807						
DB	INITIAL	806	818	833	847	861	874	888	902
DC	INITIAL	817	917						
DCA	INITIAL	832							
DCB	INITIAL	846							
DCC	INITIAL	860							
DCD	INITIAL	873							
DCE	INITIAL	887							
DCF	INITIAL	901							
DCL	MUL-DRILL1	672	671	673					
DD	INITIAL	916	936	952	964	978	990		
DE	INITIAL	935							
DECLARAT	DCL-LINEAR	259	101	281	303	483	504	528	661
			674	763	1095	2514			
DF	INITIAL	951	1002	1015	1027				
DFA	INITIAL	963							
DFB	INITIAL	977							
DFC	INITIAL	989							
DG	INITIAL-DR	1001	1037						
DGA	INITIAL-DR	1014							

UNIT	BLOCK	UNIT LOCATION	REFERENCES TO UNIT						
DGA	INITIAL-DR	1026							
DM	INITIAL-DR	1036	2607						
DISTMATR	CON-DISTMA	2398	102	2477	2404				
DONE	DRILL2-ENT	1118	1103						
DRAW	-SUBROUTIN	2611	143	199	333	349	379	421	425
			449	474	548	614	830	844	859
			871	885	899	915	961	975	987
			1000	1012	1024	1035	1052	1054	1056
			1082	1164	1166	1169	1307	1309	1342
			1344	1429	1440	1451	1497	1499	1555
			1559	1563	1436	1713	1717	1721	1808
			1858	1862	1866	1941	2180	2212	2231
			2249	2427	2464				
DRILL1	MUL-DRILL1	635	279	2605					
DRILL2	INITIAL-DR	1060	279	1094	1096	2605			
EA	DRILL2-ENT	1145	1246						
EAA	DRILL2-ENT	1172							
EAB	DRILL2-ENT	1185							
EAC	ENTIRE	1205							
EAD	ENTIRE	1226							
EB	ENTIRE	1245	1245	1280	1290				
EC	ENTIRE	1264	1302	1329					
ECA	ENTIRE	1279							
ECB	ENTIRE	1289							
ED	ENTIRE	1301	1339	1371					
EDA	ENTIRE	1328							
EE	ENTIRE	1338	2595						
EEA	ENTIRE	1370							
ENDLESS	-SUBROUTIN	2608	101						
ENDTOPIC	-SUBROUTIN	2593	258	481	434	1059	1381	1510	2031
			2136	2397					
ENTIRE	DRILL2-ENT	1121	101	1146	1173	1186	1206	1227	2604
FA	PARTS	1401							
FB	PARTS	1411	1458	1468	1479	1485			
FBA	PARTS	1423							
FBB	PARTS	1435							
FBC	PARTS	1446							
FC	PARTS	1457	1491						
FCA	PARTS	1467							
FCB	PARTS-I/O	1478							
FCC	PARTS-I/O	1484							
FD	PARTS-I/O	1490	2595						
FILL	-SUBROUTIN	2622	146	197	201	331	347	377	419
			423	447	472	1147	1170	1180	1183
			1197	1238	1556	1560	1564	1598	1620
			1639	1641	1714	1718	1722	1811	1813
			1859	1863	1867	1882	1902	1944	1946
GA	PARTS-I/O	1529	1629						
GAA	PARTS-I/O	1545							
GAB	PARTS-I/O	1566							
GAC	ARRAY I/O	1587							
GAD	ARRAY I/O	1607							
GB	ARRAY I/O	1628	1673						
GC	ARRAY I/O	1672	1696	1725	1741	1760	1777		
GD	ARRAY I/O	1695	1795						

UNIT	BLOCK	UNIT LOCATION	REFERENCES TO UNIT						
GDA	ARRAY I/O	1724							
GDB	ARRAY I/O	1740							
GDC	ARRAY I/O	1759							
GDD	ARRAY I/O	1776							
GE	ARRAY I/O	1794	1838	1850	1870	1892	1912		
GF	ARRAY I/O	1837	1927						
GFA	ARRAY I/O	1849							
GFB	ARRAY I/O	1869							
GFC	ARRAY I/O	1891							
GFD	ARRAY I/O	1911							
GG	ARRAY I/O	1926	1964	1983	1994				
GH	ARRAY I/O	1963	2000	2010	2024				
GHA	I/O=MATMUL	1982							
GHB	I/O=MATMUL	1993							
GI	I/O=MATMUL	1999	2595						
GIA	I/O=MATMUL	2009							
GIB	I/O=MATMUL	2023							
GUESS	DRILL2=ENT	1092	1091	1093	1117				
HA	I/O=MATMUL	2048							
HAA	I/O=MATMUL	2061							
HB	I/O=MATMUL	2067	2126	2595					
HBA	MATMUL=CON	2086							
HBB	MATMUL=CON	2092							
HBC	MATMUL=CON	2101							
HBD	MATMUL=CON	2114							
HBE	MATMUL=CON	2124							
HCA	LIN=MULTID	482	505						
HCB	LIN=MULTID	493	529						
HCBA	LIN=MULTID	499	498						
I	MATMUL=CON	2162							
IA	MATMUL=CON	2169	2278						
IAA	CONNMATR	2193							
IAB	CONNMATR	2206							
IAC	CONNMATR	2223							
IAD	CONNMATR	2242							
IAE	CONNMATR	2260							
IB	CONNMATR	2277	2308	2337	2344	2357	2366	2375	2384
IC	CONNMATR	2307	2387	2595					
ICA	CONNMATR	2336							
ICB	CONNMATR	2343							
ICC	CONNMATR	2356							
ICD	CONNMATR	2365							
ICE	CONNMATR	2374							
ICF	CON=DISTMA	2385							
IN	=SUBROUTIN	2624	2629						
INITIAL	DR_INITIAL	762	279	779	2405				
J	CON=DISTMA	2419							
JA	CON=DISTMA	2439							
JAB	CON=DISTMA	2451							
JAC	CON=DISTMA	2458							
JB	CON=DISTMA	2476	2493	2523	2529	2538	2548	2559	2569
			2582						
JC	DISTMATR	2492	2583	2595					
JCA	DISTMATR	2522							
JCB	DISTMATR	2528							

LESSON PLTARRAY AT 17.21.26. ON 06/13/73

UNIT	BLOCK	UNIT LOCATION	REFERENCES TO UNIT						
JCC	DISTMATR	2537							
JCD	DISTMATR	2547							
JCE	DISTMATR	2558							
JCF	DISTMATR	2568							
JCG	DISTMATR	2580							
LINEAR	DCL-LINEAR	280	279	320	338	494	2605		
LOOP	INITIAL-DR	1075	1081						
MAOUI	DR-INITIAL	731	747						
MATRMULT	I/O-MATMUL	2032	102	2068	2087	2093	2102	2115	2125
			2604						
MULTIDIM	LIN-MULTID	503	279	501	536	2605			
NEWTOPIC	-SUBROUTIN	2601							
NONE	DR-INITIAL	704	700						
NOT	DR-INITIAL	701	697	700	716	727	730	744	747
PARTS	PARTS	1383	101	1412	1424	1436	1447	2604	
START	START	32							
T2	-SUBROUTIN	2606	2595						

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-73-594	2.	3. Recipient's Accession No.

4. Title and Subtitle

ARRAYS IN PL/I
A LESSON FOR PLATO IV

5. Report Date: October 1973

6.

7. Author(s): Luis Humberto Jauregui-Topete

8. Performing Organization Rept. No. UIUCDCS-R-73-594

9. Performing Organization Name and Address
University of Illinois at Urbana-Champaign
Department of Computer Science
Urbana, Illinois 61801

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. Sponsoring Organization Name and Address
University of Illinois at Urbana-Champaign
Department of Computer Science
Urbana, Illinois 61801

13. Type of Report & Period Covered: Master's Thesis

14.

15. Supplementary Notes

16. Abstracts

This is a report on the lesson, "ARRAYS IN PL/I", taught by means of the computer-based educational system, PLATO IV, to students taking introductory computer science courses. It presents the goals and decision procedures in writing the lesson, as well as the results of a trial with students.

17. Key Words and Document Analysis. 17a. Descriptors

PLATO IV
TUTOR
arrays
computer-based education
PL/I

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
RELEASE UNLIMITED	UNCLASSIFIED	75
	20. Security Class (This Page) UNCLASSIFIED	22. Price

1973

BOUND BY
HNM

FEB 2 0 1975

UNIVERSITY OF ILLINOIS-URBANA
510.84 IL6R no. C002 no.590-594(1973
Design of totally self-checking asynchro
3 0112 088400897